

MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
SECRETARIA DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE MESTRADO EM SISTEMAS E COMPUTAÇÃO

ANDERSON CRIVELLA DE CARVALHO RODRIGUES

FUSÃO DE SENsoRES PARA UM VANT VIA SUAVIZAÇÃO
INCREMENTAL BASEADA EM GRAFOS-FATORES

Rio de Janeiro
2015

INSTITUTO MILITAR DE ENGENHARIA

ANDERSON CRIVELLA DE CARVALHO RODRIGUES

**FUSÃO DE SENSORES PARA UM VANT VIA SUAVIZAÇÃO
INCREMENTAL BASEADA EM GRAFOS-FATORES**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Sistemas e Computação.

Orientador: Prof. Paulo Fernando Ferreira Rosa - Ph.D.

Rio de Janeiro
2015

c2015

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80-Praia Vermelha
Rio de Janeiro-RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do autor e do orientador.

xxxxx Rodrigues, A. C. C.

Fusão de sensores para um VANT via suavização incremental baseada em grafos-fatores/ Anderson Crivella de Carvalho Rodrigues.

– Rio de Janeiro: Instituto Militar de Engenharia, 2015.
xxx p.: il., tab.

Dissertação (mestrado) – Instituto Militar de Engenharia – Rio de Janeiro, 2015.

1. Robôs móveis autônomos. 2. Robôs cooperativos. I. Título. II. Instituto Militar de Engenharia.

CDD 629.892

INSTITUTO MILITAR DE ENGENHARIA

ANDERSON CRIVELLA DE CARVALHO RODRIGUES

**FUSÃO DE SENsoRES PARA UM VANT VIA SUAVIZAÇÃO
INCREMENTAL BASEADA EM GRAFOS-FATOReS**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Sistemas e Computação.

Orientador: Prof. Paulo Fernando Ferreira Rosa - Ph.D.

Aprovada em 04 de fevereiro de 2015 pela seguinte Banca Examinadora:

Prof. Paulo Fernando Ferreira Rosa - Ph.D. do IME - Presidente

Prof. Max Suell Dutra - Dr. Ing., da UFRJ

Prof. Esteban Walter Gonzalez Clua - D.Sc., da UFF

Prof. Claudia Marcela Justel - D.Sc., do IME

Rio de Janeiro
2015

Dedico esta à minha esposa e à minha querida filha.

AGRADECIMENTOS

Agradeço, primeiramente, ao Supremo Criador, que sempre proporcionou-me forças, principalmente nos momentos de dúvidas, desânimo e dificuldades. Agradeço especialmente à minha esposa e família pela compreensão nos momentos de ausência e apoio nos momentos difíceis. Ao professor Paulo Rosa pela simplicidade e paciência na orientação de minha dissertação. Aos companheiros da Marinha do Brasil, em especial, à Diretoria de Sistemas de Armas da Marinha (DSAM), que confiou em meu trabalho. Aos amigos do laboratório de robótica, Alberto, Luis, Raphael, Leandro e Vidal, que compartilharam alegrias, sofrimentos, ideias e opiniões. Ao instrutor de voo Ferreirinha, que gentilmente, pilotou o VANT na condução do ensaio de voo. Por fim, a todos os professores e funcionários do Departamento de Engenharia de Sistemas (SE/8) do Instituto Militar de Engenharia.

Anderson Crivella de Carvalho Rodrigues

Sonhos determinam o que você quer. Ação determina o que você conquista.

Aldo Novak

SUMÁRIO

LISTA DE ILUSTRAÇÕES	10
LISTA DE TABELAS	15
1 INTRODUÇÃO	19
1.1 Motivação	19
1.2 Objetivos e contribuições	23
1.3 Estrutura da dissertação	23
2 REVISÃO DA LITERATURA	25
2.1 Desenvolvimento e simulação de VANT	25
2.2 Fusão de Sensores	27
3 TÓPICOS TUTORIAIS	31
3.1 Sistemas de referência	31
3.2 Sensores	33
3.2.1 Acelerômetro	34
3.2.2 Girômetro	34
3.2.3 Magnetômetro	35
3.2.4 Barômetro	36
3.2.5 Sensor de Pressão Diferencial	37
3.2.6 GPS	38
3.3 Grafos: conceitos básicos	40
3.4 Modelos Gráficos Probabilísticos	42
3.4.1 Redes Bayesianas	43
3.4.2 Grafos-fatores	43
3.5 Otimização via mínimos quadrados	44
4 FORMULAÇÃO DO PROBLEMA	47
4.1 Formulação via método dos mínimos quadrados	48
5 VEÍCULO AÉREO NÃO TRIPULADO	56
5.1 Estrutura aerodinâmica	56

5.2	Movimentos e estados da aeronave	57
5.3	Piloto automático Navstik	60
5.4	Visão esquemática	64
6	SOLUÇÃO: PRÉ-INTEGRAÇÃO E SUAVIZAÇÃO INCREMENTAL	66
6.1	Front-end: pré-integração	66
6.2	Back-end: suavização incremental	68
6.2.1	Inferência incremental	71
7	METODOLOGIA	81
7.1	Hardware-in-the-loop (HIL)	81
7.1.1	HIL via dados de sensores	82
7.1.2	HIL via dados de estados	82
7.2	Arquitetura da simulação proposta	83
7.2.1	Simulador de voo X-Plane®	83
7.2.2	Estação de Controle em Solo Qgroundcontrol	85
7.2.3	Protocolo MAVlink	86
7.2.4	Anatomia do pacote	87
7.2.4.1	Mensagem HIL_STATE_QUATERNION	88
7.2.5	Plugin de fusão sensorial	89
8	EXPERIMENTOS E RESULTADOS	92
8.1	Experimentos	92
8.1.1	Voo quadrado	92
8.1.2	Voo montanha-russa	99
8.1.3	Voo <i>lawn-mower</i>	105
8.2	Resultados consolidados das simulações	111
8.3	Ensaio de voo	111
8.3.1	Análise	112
9	CONSIDERAÇÕES FINAIS	115
9.1	Trabalhos futuros	116
10	REFERÊNCIAS BIBLIOGRÁFICAS	118

11 APÊNDICES	126
11.1 Apêndice A: Otimização via Mínimos Quadrados	126
11.1.1 Problemas lineares	126
11.1.2 Problemas lineares ponderados	130
11.1.3 Problemas não-lineares	132
11.1.4 Problemas não-lineares ponderados	134

LISTA DE ILUSTRAÇÕES

<p>FIG.1.1 O <i>front-end</i> processa os dados dos sensores e do movimento e constrói o grafo-fator. O <i>back-end</i> obtém uma solução aplicando o método da otimização incremental.</p> <p>FIG.2.1 Estação de controle em solo capaz de controlar o voo e a navegação de múltiplos VANTs em ambiente simulado. (ANGONESE, 2013a)</p> <p>FIG.3.1 Sistemas de referência ECI, ECEF, NED e do corpo. Adaptado de (FOSSEN, 2011)</p> <p>FIG.3.2 Referencial do simulador em comparação com o referencial do corpo da aeronave.</p> <p>FIG.3.3 Representação conceitual de um acelerômetro e um girômetro. Adaptado de (DAPAIKÃO, 2011)</p> <p>FIG.3.4 Representação conceitual da medição do campo magnético.</p> <p>FIG.3.5 Relação entre a variação da altura com a variação da pressão.</p> <p>FIG.3.6 Representação esquemática do funcionamento do sensor de pressão diferencial. Adaptado de (SYSTEMS, 2014).</p> <p>FIG.3.7 Representação esquemática da determinação da posição GPS em 3D. Caso sem erros de sincronização satélite-receptor. (KAPLAN, 2005).</p> <p>FIG.3.8 Representação esquemática da determinação da posição GPS em 3D. Caso real com erros de sincronização satélite-receptor. (KAPLAN, 2005).</p> <p>FIG.3.9 Um grafo dirigido G e o subgrafo induzido por W.</p> <p>FIG.3.10 Um grafo bipartido. Os vértices V estão particionados nos subconjuntos $U = \{\text{vértices pretos}\}$ e $W = \{\text{vértices brancos}\}$</p> <p>FIG.3.11 Uma árvore com o nó raiz v_1. O nó v_2 é filho de v_1 e pai dos nós v_3, v_4 e v_5.</p> <p>FIG.3.12 Rede bayesiana representando a função de probabilidade conjunta $P(x_1, x_2, x_3, x_4, x_5, x_6) = P(x_1 x_2)P(x_2 x_3)P(x_3 x_4)P(x_4)P(x_5 x_1, x_2)P(x_6 x_1, x_3, x_4)$.</p>	<p>22</p> <p>26</p> <p>32</p> <p>33</p> <p>35</p> <p>36</p> <p>37</p> <p>38</p> <p>39</p> <p>40</p> <p>41</p> <p>41</p> <p>42</p> <p>43</p>
--	---

FIG.3.13	Grafo-fator para a função $f(x_1, x_2, x_3)$.	44
FIG.4.1	Grafo-fator simplificado ilustrando os fatores que associam duas posições consecutivas (modelo do movimento, em azul) e os fatores que associam um estado com as medições (modelo das medições, em verde).	50
FIG.4.2	Grafo-fator para o exemplo. Ele representa o movimento de uma aeronave no plano \mathbb{R}^1 .	52
FIG.4.3	Representação das funções de densidade de probabilidade das posições da aeronave.	55
FIG.4.4	Tempo de processamento em função da quantidade de fatores. O processamento incremental possui complexidade de tempo constante enquanto que a solução batch cresce linearmente com a quantidade de fatores.	55
FIG.5.1	VANT de asa fixa em formato delta, suas dimensões e seu centro de gravidade.	57
FIG.5.2	Superfícies de controle de uma aeronave. Adaptado de (COOK, 2012)	59
FIG.5.3	Representação esquemática dos eixos de rotação e translação do VANT.	59
FIG.5.4	Movimentos realizados pelo VANT em função dos elevons.	60
FIG.5.5	Navstik com seu microcontrolador cortex M4 e os sensores.	61
FIG.5.6	Visão artística interna do projeto microVANT do laboratório de Robótica do IME. Fonte: Alberto Angonese e Victor Betio.	62
FIG.5.7	Placa de expansão IvyPro. Responsável por prover as entradas e saídas para a placa Navstik.	63
FIG.5.8	Modelo esquemático dos componentes do VANT.	65
FIG.6.1	Grafo-fator sem (a) e com pre-integração. A técnica diminui a quantidade de nós adicionados ao grafo.	67
FIG.6.2	(a) O grafo-fator e sua matriz Jacobiana A para um exemplo hipotético. (b) A rede Bayesiana triangulada e a matriz de informação R resultante da ordem de eliminação l_1, l_2, x_1, x_2, x_3 . (c) A árvore de Bayes e seus cliques (KAESS, 2011a).	70

FIG.6.3	Atualização da árvore Bayesiana da figura 6.2(c) quando um novo fator entre x_1 e x_3 (em azul) é adicionado. A subárvore da direita não é afetada. (KAESS, 2011a)	73
FIG.6.4	Representação ilustrativa do grafo de navegação do VANT. Os sensores (nós coloridos) operam com taxas diferentes e vão sendo adicionados ao grafo, onde f^{pd} é o fator do sensor de pressão diferencial, f^{baro} é o fator do barômetro, f^{GPS} é o fator do GPS e f^{IMU} é o fator do movimento.	73
FIG.6.5	Grafo-fator para o EX. 6.1. No instante $t = 4s$ recebemos uma nova medição. As partes afetadas estão tracejadas.	74
FIG.6.6	Grafo-fator após a eliminação da variável x_0 . Trata-se de um modelo híbrido entre um grafo-fator com uma rede Bayesiana. O novo fator resultante da eliminação está representado em amarelo.	76
FIG.6.7	Grafo híbrido após a eliminação da variável x_1 . O novo fator resultante da eliminação está representado em amarelo.	77
FIG.6.8	Grafo híbrido após a eliminação da variável x_2 . O novo fator resultante da eliminação está representado em amarelo.	78
FIG.6.9	Rede Bayesiana resultante da eliminação de todas as variáveis do grafo-fator do EX. 6.1. A probabilidade conjunta é $P(X) = P(x_0 x_1)P(x_1 x_2)P(x_2 x_3)P(x_3)$	79
FIG.6.10	Árvore Bayesiana resultante da aplicação do Algoritmo 4 na rede de Bayes da FIG. 6.9.	79
FIG.6.11	Grafo afetado pela nova medição do tempo $t = 4$ e as correspondentes rede e árvore Bayesianas.	80
FIG.6.12	Nova árvore Bayesiana após substituir o clique C_{raiz} anterior pela árvore da FIG. 6.11c	80
FIG.7.1	HIL via dados de sensores. Os dados dos sensores são enviados para o piloto automático que realiza a fusão dos sensores.	82
FIG.7.2	HIL via dados de estados. Os dados dos sensores são enviados para um componente externo que irá realizar a fusão de sensores. O piloto automático recebe as estimativas prontas.	83
FIG.7.3	Arquitetura de simulação. O plugin recebe os dados dos sensores provenientes do simulador de voo, adiciona ruído branco, gera o	

grafo-fator e finalmente, executa a suavização incremental. O protocolo MAVLink envia as estimativas para a ECS e o piloto automático que retorna as saídas do controle da atitude e posição.	84
FIG.7.4 Visão do operador durante a realização de uma missão pelo QGroundControl.	86
FIG.7.5 Representação esquemática de um pacote MAVLink. (MEIER, 2011)	87
FIG.7.6 Tela do plugin de fusão sensorial.	90
FIG.8.1 Padrão quadrado (preto) e as trajetórias real (azul), incremental (vermelho) e batch (amarelo).	94
FIG.8.2 Voo quadrado. Estimativas para a orientação.	95
FIG.8.3 Voo quadrado. Estimativas para a posição.	96
FIG.8.4 Voo quadrado. Estimativas para a velocidade.	97
FIG.8.5 Voo quadrado. Erros absolutos da orientação, posição e velocidade.	98
FIG.8.6 Padrão montanha-russa (preto) e as trajetórias real (azul), incremental (vermelho) e batch (amarelo).	100
FIG.8.7 Voo montanha russa. Estimativas para a orientação.	101
FIG.8.8 Voo montanha russa. Estimativas para a posição.	102
FIG.8.9 Voo montanha russa. Estimativas para a velocidade.	103
FIG.8.10 Voo montanha russa. Erros absolutos da orientação, posição e velocidade.	104
FIG.8.11 Padrão lawn-mower (preto) e as trajetórias real (azul), incremental (vermelho) e batch (amarelo).	106
FIG.8.12 Voo lawn-mower. Estimativas para a orientação.	107
FIG.8.13 Voo lawn-mower. Estimativas para a posição.	108
FIG.8.14 Voo lawn-mower. Estimativas para a velocidade.	109
FIG.8.15 Voo lawn-mower. Erros absolutos da orientação, posição e velocidade.	110
FIG.8.16 Preparando a eletrônica embarcada para os voos manual e autônomo.	
112	
FIG.8.17 Crescimento no erro do ângulo de rolamento quando não há auxílio do GPS.	112

FIG.8.18 Ensaio de voo com aconchamento da missão via estação de controle em solo.	113
FIG.8.19 Comparativo entre o voo real e o simulado.	114
FIG.11.1 Resultado da estimativa do EX. 11.1.	130
FIG.11.2 Resultado da estimativa do EX. 11.2.	133

LISTA DE TABELAS

TAB.4.1	Medições provenientes da odometria e dos sensores.	52
TAB.4.2	Tempos de processamento para as soluções batch e incremetal.	55
TAB.5.1	Variáveis de estado necessárias para a localização do VANT.	58
TAB.7.1	Descrição do pacote MAVLink	88
TAB.7.2	Estrutura da mensagem HIL_STATE_QUATERNION enviada para o piloto automático.	89
TAB.8.1	Taxas de operação e ruídos dos sensores utilizados durante os experimentos.	92
TAB.8.2	Resultados.	111
TAB.8.3	Dados estatísticos da altitude (definida como 100 m) do voo real e simulado.	114
TAB.11.1	Tabela com a posição real $y(t)$, as medições do radar $y_r(t)$ e o erro $e(t)$ associado às 10 medições realizadas por um radar com $\sigma = 3m$ e $f = 1Hz$	128
TAB.11.2	Tabela com a posição real $y(t)$, as medições $y_r(t)$ e o erro $e(t)$ associado às 10 medições realizadas por dois radares (O primeiro com $\sigma = 3m$ e $f = 1Hz$ e o outro com $\sigma = 0,5m$ e $f = \frac{1}{3}Hz$).	132

LISTA DE ALGORITMOS

1	Pré-integração das medições da IMU $[f_j, w_j]$. (INDELMAN, 2013).	67
2	Predição do fator IMU. (INDELMAN, 2013).....	68
3	(Grafo-fator para rede Bayesiana) Transforma um grafo-fator em uma rede Bayesiana mediante a eliminação de todas as variáveis x_j dada uma ordem de eliminação. Adaptado de (KAESS, 2011a)	69
4	(Rede Bayesiana para árvore Bayesiana). Gera uma árvore Bayesiana a partir de uma rede Bayesiana triangulada. Adaptado de (KAESS, 2011a).	71
5	(Atualização da árvore Bayesiana). Atualiza a árvore com os novos fatores F_{novo} . Adaptado de (KAESS, 2011a)	72
6	Plugin de fusão sensorial. Obtém os dados dos sensores do simulador e calcula as estimativas incrementais para os estados do VANT.....	91

RESUMO

Diversos problemas da robótica e da visão computacional como *simultaneous localization and mapping* (SLAM), *bundle adjustment* (BA) e navegação podem ser formulados como a otimização, via método dos mínimos quadrados, de funções resíduo representadas por um grafo. Nestas abordagens, cada movimento do robô ou cada medição realizada por um sensor é representada como um nó em um grafo. Arestras representam restrições entre esses nós. A primeira parte do problema, conhecida como *front-end*, é a construção desse grafo. A segunda parte, conhecida como *back-end*, objetiva encontrar a configuração que melhor explica as restrições modeladas pelas funções resíduo. O trabalho descreve a estrutura geral de tais problemas objetivando a fusão dos sensores de um VANT - sistema robótico aéreo de dimensões reduzidas, baixo custo e caráter dual.

Para resolver o problema, o trabalho foi dividido em duas etapas: (a) a elaboração e a montagem do VANT e (b) o desenvolvimento da fusão dos sensores. Na primeira etapa, foi escolhido um modelo aerodinâmico de asa fixa com formato delta, por ser barato, largamente comercializado e de fácil emprego por simuladores de voo. Completa a solução, um conjunto de sensores composto por uma unidade de medida inercial, um magnetômetro, um barômetro, um sensor de pressão diferencial e um GPS. Esses sensores tipicamente operam com diferentes taxas ($1 \sim 100\text{Hz}$) e são contaminados por ruídos não determinísticos, prejudicando a qualidade das informações e fazendo com que o grafo de navegação cresça rapidamente. Na etapa da fusão de sensores, o trabalho emprega como *back-end* um método inovador chamado de suavização incremental (*incremental smoothing*) via grafos-fatores (*factor-graphs*), o *front-end*. Esse método otimiza apenas um pequeno subconjunto dos nós do grafo reduzindo consideravelmente a carga computacional. O subconjunto representa os nós que são afetados toda vez que uma nova medição sensorial é incorporada.

Como metodologia de validação propõe-se um teste de HIL (hardware-in-the-loop) composto do simulador de voo X-Plane®, de uma estação de controle em solo e do hardware que contém as funcionalidades de piloto automático. O simulador de voo foi estendido com o modelo aerodinâmico da aeronave real e com o plugin responsável pela fusão dos sensores. Para a comunicação entre os módulos da simulação adotou-se o protocolo MAVLink.

Além disso, elaborou-se três experimentos caracterizando missões de voo a serem cumpridas pelo VANT: um voo quadrado, um voo montanha-russa e um voo *lawn-mower* (útil para busca e varredura de áreas). Finalmente, um experimento real é realizado para verificar a similaridade dos resultados.

ABSTRACT

Many robotics and computer vision problems such as simultaneous localization and mapping (SLAM), bundle adjustment (BA) and navigation can be formulated as a least squares optimization of error functions represented by a graph. In these strategies, each robot movement or each sensor measurement is represented as a node in a graph. Edges represent constraints between these nodes. The first part of the problem, known as front-end is building the graph. The second part, known as back-end, aims to find a configuration that best explains constraints modeled as error functions. This work describes general structure of such problems aiming at sensor fusion of a UAV - a small size, low cost, and dual-use aerial robotic system.

In order to solve the problem, this work is divided into two stages: (a) preparation and assembly of the UAV and (b) development of sensor fusion. In first stage, a delta fixed wing aerodynamic model was chosen because it is cheap, easy to find in market and easily adopted by flight simulators. Besides, this solution encompasses a set of sensors comprised by an inertial measurement unit, a magnetometer, a barometer, a differential pressure sensor and a GPS. These sensors typically operate at different rates ($1 \sim 100$ Hz) and are corrupted by non-deterministic noises, degrading the quality of information and making navigation graph grow quickly. In sensor fusion stage, we use incremental smoothing via factor-graphs. This method optimizes only a small subset of the graph nodes, reducing computational load considerably. The subset represents nodes that are affected every time a new sensor measurement is incorporated.

Furthermore, in order to validate the methodology, we propose an HIL (hardware-in-the-loop) test composed of the X-Plane flight simulator, the ground control station and hardware that contains autopilot features. We extended the flight simulator with aerodynamic model of the actual aircraft and sensor fusion plugin. For communication between simulation modules, we adopted MAVLink protocol.

In addition, we have settled three experiments featuring flight missions to be met by UAV: a square flight, a roller-coaster flight and lawn-mower flight (useful for rescue and area coverages). Finally, an actual experiment is conducted to determine the similarity of results.

1 INTRODUÇÃO

Os esforços no desenvolvimento de VANTs têm aumentado consideravelmente nos últimos anos, motivado pelo potencial das suas aplicações e pela crescente capacidade de miniaturização dos equipamentos eletrônicos. As vantagens de um VANT incluem: um sistema compacto, capacidade de aquisição de dados em tempo real, uma baixa seção-reta radar (*radar cross section* - medida da habilidade de um alvo refletir os sinais do radar na direção do receptor), uma baixa emissão de ruído, um baixo custo de produção e a capacidade de realizar voos de baixa altitude. Estas características tornam-os ideais para missões de reconhecimento, varredura de áreas e detecção de agentes biológicos ou materiais nucleares. Contudo, o sucesso dessas missões está diretamente relacionado com a qualidade da localização e da navegação. O núcleo desse processo fundamental encontra-se na aquisição de dados físicos e atmosféricos diversos que possam contribuir de alguma forma no cálculo da posição e orientação (pose) da aeronave. Esse papel cabe aos sensores. No entanto, cada sensor possui características distintas dos outros, como o nível de ruído e a velocidade ou taxa de operação. A união das características e informações de cada sensor é o que chamamos de fusão de sensores, onde o objetivo é a busca da melhor precisão possível. Do exposto, fica claro que sem os sensores com seus dados processados corretamente, o piloto automático não pode operar, ou seja, a navegação autônoma não é possível. É no procedimento fundamental da fusão dos sensores que o trabalho está fundamentado.

1.1 MOTIVAÇÃO

Mapeamento, inspeção, pesquisa atmosférica, e até mesmo fotografias aéreas são algumas das muitas áreas que começaram a empregar veículos aéreos não tripulados (VANT) - aeronaves que não necessitam de pilotos em seu interior (CAI, 2010; REUDER, 2008; WANG, 2010; HATTENBERGER, 2013; WINKVIST, 2013). Com isso, dois modos de operação são possíveis: (a) manual e (b) automático. O primeiro modo é aquele onde existe um piloto em solo responsável pela operação segura da aeronave. O último modo é aquele que não necessita de operadores, podendo o VANT voar e executar missões sozinho. Na prática, um VANT opera alternando os dois modos, onde as tarefas de decolagem e

pouso são realizadas por um operador e após a estabilização do voo, o modo automático pode ser acionado. Nesse modo, uma gama de aplicações, como as repetitivas e perigosas, podem ser realizadas, pois não há a presença do ser humano (piloto embarcado). É o caso das áreas afetadas por catástrofes nucleares ou por desastres naturais como os terremotos, inundações ou vulcões (CALTABIANO, 2005; SUZUKI, 2008; PÖLLÄNEN, 2009). (NETO, 2011) relata o emprego de um VANT na operação de monitoramento do desastre ocorrido na região serrana do Rio de Janeiro em 2010. O desastre foi tão intenso, que modificou a topografia da região e ocasionou grande número de vítimas.

Além disso, observa-se um esforço governamental para a liberação de fomentos incentivando pesquisas para a melhoria da qualidade das tecnologias relacionadas à VANTs. Dentre estes, a Chamada Pública CT-AERO - VANT do MCT/FINEP (FINEP, 2009), visa apoiar projetos relacionados exclusivamente a micro e miniVANTs. Para a categoria mini, os requisitos são: arremesso com as mãos, peso inferior a 5 kg, dimensão linear máxima de 3,0 m, alcance mínimo de 20 km, autonomia de vôo mínima de 60 minutos e capaz de ser transportado, montado e operado por uma equipe de apenas duas pessoas. Para a categoria micro: arremesso com as mãos, peso máximo de 200g, dimensão linear máxima de 200 mm, alcance mínimo de 10km, autonomia de vôo mínima de 60 minutos, propulsão por motor elétrico e alimentação elétrica a bateria ou célula de combustível. De maneira mais específica, o objetivo maior desta iniciativa governamental é apoiar o desenvolvimento dos seguintes componentes de um VANT:

- **Sistemas de Navegação** e Controle;
- Sensores de reconhecimento, como câmeras óticas e/ou de infravermelho;
- **Sistemas e sensores para posicionamento**, decolagem e pouso automáticos;
- Sistemas de detecção de falhas e reconfiguração;
- Sistemas que permitam o vôo colaborativo de múltiplos veículos, possibilitando maior autonomia em missões conjuntas;
- Sistemas de armazenamento de dados e telemetria.

O presente trabalho se enquadra como elemento de pesquisa em sistemas de navegação e em sistemas e sensores para posicionamento, destacados em negrito.

Outra crescente preocupação governamental diz respeito às questões relacionadas à Defesa e à Soberania Nacional contidas na Estratégia Nacional de Defesa (END). Nesse importante documento constam diretrizes que se referem ao desenvolvimento das capacidades de monitorar e controlar o espaço aéreo, o território e as águas jurisdicionais brasileiras. Tal desenvolvimento dar-se-á a partir da utilização de tecnologias de monitoramento terrestre, marítimo, aéreo e espacial que estejam sob inteiro e incondicional domínio nacional (MINISTÉRIO DA DEFESA, 2008). Contexto onde os VANTs e o trabalho se enquadram perfeitamente.

No entanto, essa crescente gama de aplicações e projetos de pesquisas civis e militares só foi possível devido ao surgimento de baterias de alta potência, dispositivos de rádio de baixo consumo e longo alcance, fuselagens leves e baratas, poderosos microprocessadores e mais especificamente dos sensores de tecnologia MEMS (*Microelectromechanical systems*). Sensores são dispositivos que interagem diretamente com o ambiente e que são em última instância, a fonte de todos os dados de entrada em um sistema de fusão de múltiplos sensores (FOWLER, 2004). Já a tecnologia MEMS é o uso de elementos mecânicos, eletrônicos ou eletromecânicos miniaturizados utilizando as técnicas de microfabricação. Esses diversos fatores possibilitaram uma drástica redução dos custos, consequentemente, viabilizando a maior disseminação dos VANTs. Contudo, no caso particular dos sensores, dado o alto grau de miniaturização, uma característica importante foi prejudicada: a precisão da localização.

A comunidade robótica enfrenta o problema da localização há algumas décadas utilizando diversos tipos de algoritmos e sensores. São utilizados desde os conhecidos sensores de posicionamento global e unidades de medidas iniciais até as câmeras de visão computacional e sensores que utilizam o laser como mecanismo de operação. Nesse contexto, é de grande interesse encontrar uma solução probabilística que possa melhorar a precisão ou minimizar os erros da localização. Uma dessas soluções é a fusão de sensores: metodologia utilizada para a combinação de dados de diversos sensores com o objetivo de melhorar a qualidade das informações, de modo que ela seja, de algum modo, melhor do que se os sensores fossem utilizados individualmente (MITCHELL, 2012). Diversas abordagens diferentes para a fusão de sensores podem ser encontradas na literatura (CRASSIDIS, 2007; SMITH, 2006), contudo a abordagem mais comum é a baseada em variantes do filtro de Kalman (KALMAN, 1960), notadamente o filtro de Kalman estendido (EKF). Tais filtros fornecem uma solução de tempo constante para a estimativa mais recente do

estado \mathbf{X}_t , e, portanto, podem processar dados de sensores que operam em altas taxas, em particular os sensores inerciais. Todavia, apesar do EKF ser a solução mais comum em diversas aplicações, a incorporação de medições assíncronas de diferentes sensores que operam em múltiplas frequências não é trivial. Além disso, se o sistema for altamente não-linear, as variantes do filtro de Kalman são conhecidas por serem propensas a erros. Assim, ao invés de considerar somente a estimativa mais recente do estado \mathbf{X}_t (filtragem), passou-se a considerar todas as medições anteriores para obter a estimativa do estado $\mathbf{X}_{1:t}$ (*smoothing*). Infelizmente, essa abordagem leva a um rápido crescimento das informações a serem calculadas a cada passo de tempo. Portanto, essa técnica foi quase sempre utilizada *offline*.

Não obstante, os recentes avanços no campo da suavização incremental (*incremental smoothing*) levaram a uma solução chamada iSAM2 (KAESS, 2012). Essa solução evita que se calcule novamente todas as variáveis de estado sempre que uma nova informação sensorial é incorporada, servindo como o *back-end*. A técnica utiliza procedimentos de otimização que englobam o método dos mínimos quadrados, ferramentas da álgebra linear esparsa, teoria dos grafos e modelos gráficos probabilísticos. Desses, o modelo gráfico conhecido como grafo-fator (grafo bipartido que representa fatorações de funções, possibilitando cálculos eficientes, como as marginalizações ou estimativas máxima a posteriori) é o utilizado para a construção do grafo de navegação. Serve dessa forma como o *front-end*. Nesse sentido, o trabalho propõe o emprego da suavização incremental como forma de resolver o problema de localização de um VANT conforme ilustrado pelo procedimento de alto nível da FIG. 1.1. Os dados dos sensores e do movimento são coletados e proces-

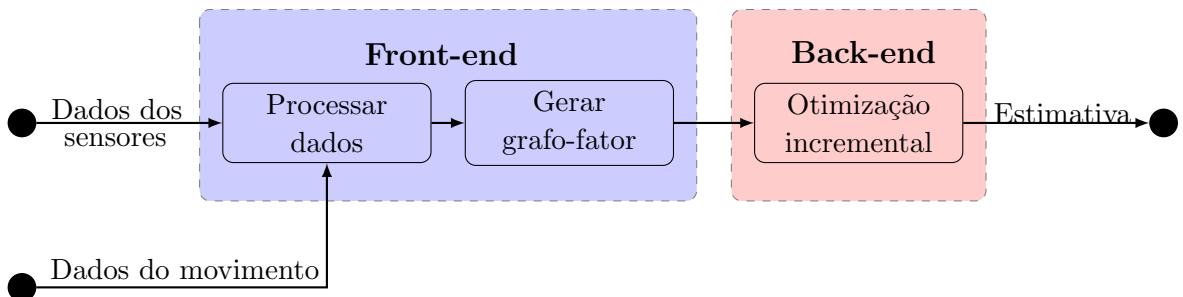


FIG. 1.1: O *front-end* processa os dados dos sensores e do movimento e constrói o grafo-fator. O *back-end* obtém uma solução aplicando o método da otimização incremental.

sados gerando nós que vão sendo adicionados ao grafo. A construção desse grafo é o que chamamos de *front-end*. À medida que o grafo cresce pela incorporação dos dados dos

sensores, o cálculo da estimativa é realizado pelo *back-end* em tempo real.

1.2 OBJETIVOS E CONTRIBUIÇÕES

O objetivo do trabalho descrito nesta dissertação é a estruturação e a fusão dos sensores de um VANT de asa fixa, mediante o emprego da técnica de suavização incremental via grafos-fatores, o que permite a criação do grafo de navegação pela simples adição de nós, possibilitando a localização no espaço ao longo do tempo. Nesse sentido, as principais contribuições são:

- a estruturação de um VANT de asa fixa, com carga útil reduzida e funcionalidades de voo autônomo, objetivando o emprego em futuras plataformas de menor porte, como os microVANT;
- a aplicação da técnica de suavização incremental como mecanismo de fusão sensorial em conjunto com as facilidades do modelo gráfico probabilístico conhecido como grafo-fator;
- a criação de uma metodologia de simulação com as características da aeronave estruturada e o desenvolvimento de um plugin que integra a fusão dos sensores com o simulador de voo, possibilitando testes mais próximos da realidade;

1.3 ESTRUTURA DA DISSERTAÇÃO

Além da introdução, a dissertação está organizada nos seguintes capítulos:

- Capítulo 2 - Revisão da Literatura: Nesse capítulo são apresentados os trabalhos que foram utilizados para o desenvolvimento da dissertação, sendo selecionados trabalhos no campo da fusão de sensores, veículos aéreos não tripulados e arquiteturas de simulações de voo.
- Capítulo 3 - Tópicos Tutoriais: Aqui são abordados os conceitos utilizados como referencial teórico para o desenvolvimento do trabalho. Os assuntos envolvem temas como: sistemas de referência, sensores, grafos e otimização.
- Capítulo 4 - Formulação do Problema: Nesse capítulo é descrito matematicamente o problema da localização de uma aeronave como uma otimização de mínimos quadrados.

- Capítulo 5 - Veículo Aéreo Não Tripulado: Nesse capítulo é descrita a estruturação do VANT, em conjunto com seus componentes e sensores.
- Capítulo 6 - Solução: pré-integração e suavização incremental: Nessa parte da dissertação apresentamos a suavização incremental e seus algoritmos e a técnica da pré-integração como forma de reduzir o tamanho do grafo de navegação.
- Capítulo 7 - Metodologia: Nesse capítulo descrevemos a arquitetura elaborada para a simulação do voo, desde a escolha dos softwares até as tecnologias acessórias.
- Capítulo 8 - Experimentos e resultados: Nesse capítulo são apresentados os resultados obtidos com a aplicação da solução proposta.
- Capítulo 9 - Considerações Finais: Finalmente, no último capítulo, teceremos nossas conclusões e apresentaremos as dificuldades encontradas em conjunto com algumas sugestões para trabalhos futuros.

2 REVISÃO DA LITERATURA

Diversas disciplinas são necessárias para o desenvolvimento completo de um VANT, onde uma série de problemas precisam ser enfrentados. Neste capítulo apresentamos os trabalhos que foram utilizados como auxílio e embasamento teórico para a pesquisa proposta, em especial aqueles relacionados à fusão de sensores com aspectos de otimização e os que tratam de maneira geral os VANTs e ambientes simulados.

2.1 DESENVOLVIMENTO E SIMULAÇÃO DE VANT

Apesar de nosso trabalho concentrar-se nos aspectos da fusão sensorial, foi necessssário o estudo e o conhecimento das operações e funcionalidades gerais de um VANT de asa fixa. Não era nossa intenção modelar aerodinamicamente um novo VANT, mas sim, prototiparmos, de uma maneira rápida, uma aeronave para os experimentos e simulações. Para uma introdução aos desafios aerodinâmicos de aeronaves de asa fixa remetemos o leitor à (MUELLER, 2003; GUERRERO, 2014). Já o trabalho de (BRONZ, 2013b) propõe o desenvolvimento de um VANT de 1 metro de envergadura com capacidade de voo entre 2 a 8 horas (com o auxílio de células solares). Apesar do voo de longa duração, a aeronave não é autônoma. Outro aspecto levado em consideração na escolha dos componentes eletrônicos foi o tamanho e o peso. Dessa forma, não haveria restrições quanto à aplicação nos futuros projetos de microVANTS do IME. O trabalho de (BRONZ, 2013a) apresenta os principais itens a serem considerados em um projeto de microVANTS: motores, hélices, tipo de asa, sensores, testes e ajustes. Com relação aos trabalhos desenvolvidos neste Instituto citamos o de (ALVES, 2010) que descreveu a modelagem e o controle de uma aeronave para utilização em aplicações de planejamento de missão de um VANT de asa fixa, e o de (ANGONESE, 2013a) que desenvolve um sistema computacional para uma ECS (estaçao de controle em solo) capaz de controlar o voo e a navegação de múltiplos VANTS (FIG. 2.1), implementando funcionalidades de voo em formação e desvio de outras aeronaves em rota de colisão. Ambos os trabalhos utilizam como aeronave, o modelo Cesnna (aeronave comercial de maior porte). Apesar de não corresponder ao nosso modelo, uma parte fundamental desse trabalho foi com relação ao uso do simulador de voo X-Plane®. Diversos detalhes operacionais, configurações e ajustes puderam ser compreendidos mais



FIG. 2.1: Estação de controle em solo capaz de controlar o voo e a navegação de múltiplos VANTs em ambiente simulado. (ANGONESE, 2013a)

rapidamente, o que nos facilitou a implementação das nossas funcionalidades. O trabalho apresenta também uma introdução sobre diversos pilotos automáticos. Contudo, mesmo capaz de controlar o voo e a navegação da aeronave, percebemos que necessitariámos de uma solução mais próxima do nosso modelo de aeronave, hardware e requisitos de comunicações para realizar as simulações. Um dos motivos foi a utilização do conceito de campos potenciais para atrair as aeronaves para um determinado *waypoint* e ao mesmo tempo repelí-las para evitar a colisão. Dessa forma era calculado o ângulo de guinada entre a aeronave e o *waypoint* representando a correção a ser feita na rota. Somente essa informação era enviada ao piloto automático do simulador de voo (além de parâmetros como a altitude e velocidade constantes). Em resumo, o controle era efetuado pelo X-Plane® e a ECS indicava para onde a aeronave deveria navegar.

Dentre os trabalhos que utilizam o X-Plane® como simulador citamos (CARVALHO, 2012; DOSSANTOS, 2011; RIBEIRO, 2010). Todos eles utilizam o mecanismo interno de empacotamento e envio de informações via UDP. Diferentemente, adotamos o protocolo MAVLink como forma de comunicação entre os componentes de simulação (ou reais). O protocolo é otimizado para aplicações robóticas, independe da camada de transporte (UDP, XBee ou porta serial, por exemplo) e é adotado por diversos pilotos automáticos. Dessa forma, tanto as simulações quanto os voos reais podem utilizar a mesma infraestrutura de comunicações. Os detalhes do protocolo encontram-se no Capítulo 7.

2.2 FUSÃO DE SENsoRES

Nas últimas duas décadas, os sistemas robóticos autônomos têm sido utilizados numa variedade de aplicações de pesquisa, sejam para uso civil ou militar, notadamente a navegação de robôs terrestres ou de veículos aéreos não tripulados. Navegação precisa e confiável é um requisito fundamental em tais sistemas e tem sido o foco de muitos esforços recentes de pesquisa. Os principais componentes nessa engrenagem são os sensores. Enquanto que nos primórdios dos sistemas de navegação, a unidade de medida inercial foi o sensor principal, os sistemas atuais possuem capacidades de detecção adicionais, tais como o sistema de posicionamento global (GPS), magnetômetros, câmeras monoculares e câmeras estéreo e sensores de alcance e altitude. Calcular uma solução de navegação torna-se, então, uma questão de fusão de sensores. O que torna o problema ainda mais complexo é a necessidade de se produzir uma estimativa em tempo real. Se considerarmos ainda as restrições de peso e dimensão impostas ao desenvolvimento de um VANT, os desafios são ainda maiores. Uma abordagem para a fusão de sensores é transformar o problema no cálculo de uma estimativa máximo a posteriori (MAP) dos estados do sistema dado todas as medições disponíveis. A solução ideal envolve a execução de uma otimização *batch* ou *offline* após a obtenção de todas as medições recebidas. Esta abordagem, também conhecida como *bundle adjustment* (BA), é bastante utilizada na comunidade robótica para resolver o problema conhecido como SLAM (*Simultaneous Localization and Mapping*), onde o mapa do ambiente pode ser construído após o processamento dos dados (THRUN, 2005; DELLAERT, 2006; FOLKESSON, 2005; EUSTICE, 2006). Recentemente, a otimização *batch* foi aplicada para a fusão de sensores em sistemas de navegação iniciais (MOURIKIS, 2007, 2008; BRYSON, 2009). No entanto, nessa abordagem, a cada nova medição, deve-se recalcular novamente todas as variáveis recebidas até o momento. A solução torna-se inviável para sistemas de tempo real, uma vez que as informações a serem recalculadas crescem rapidamente com o tempo. Assim, os sistemas descritos em (MOURIKIS, 2007, 2008; BRYSON, 2009) apenas realizam otimizações *batch* periodicamente ou completamente *offline*.

Nesse sentido, para obter-se desempenho, muitas implementações usam variantes do filtro de Kalman estendido (EKF). Este filtro obtém desempenho de tempo real por marginalizar todos os estados passados e estimar apenas o estado atual. A abordagem tem sido utilizada em uma variedade de aplicações, tais como a fusão de sensores de um veículo

submarino (SKOGLUND, 2012), rastreio de alvos (NORDSJO, 2005) e veículos aéreos não tripulados (KINGSTON, 2004; YONGLIANG, 2008; MAGNUSSON, 2013; MAO, 2007). Enquanto o filtro de Kalman fornece uma solução ideal no caso linear (FARRELL, 2008), a maioria dos modelos de sensores incluem não linearidades. O EKF, e até mesmo algumas variantes como o EKF iterado que suportam relinearização, são incapazes de realizar uma adequada relinearização quando os estados passados não fazem mais parte do filtro, uma vez que foram marginalizados. O EKF de estado aumentado e os suavizadores de janela fixa superam parcialmente este problema, mantendo parte dos estados passados (até o tamanho fixado). Por exemplo, um EKF de estado aumentado é usado em (MOURIKIS, 2007) para incorporar as informações fornecidas por múltiplas observações de características visuais em uma solução de navegação autônoma. Em (TARDIF, 2010), o EKF de estado aumentado é aplicado para fundir odometria visual com navegação inercial, enquanto (SIBLEY, 2010) usa suavização de janela fixa para a reconstrução 3D de uma superfície planetária com base em imagens estéreo que são adquiridas por uma plataforma autônoma. No entanto, o aumento do tamanho do estado tem um grande impacto sobre a complexidade computacional. Conforme já comentado, algumas técnicas atuais exigem a atualização de todas as variáveis ou daquelas dentro da janela de atraso sempre que uma medição é incorporada. Isso quer dizer que, a fusão de sensores usando suavizadores de janela fixa é subótima, aproximando-se da estimativa MAP apenas para janelas suficientemente grandes. Além disso, tal como descrito em (HUANG, 2011), filtros e suavizadores de janela fixa podem tornar-se probabilisticamente inconsistentes quando as variáveis marginais são linearizadas em torno de um ponto de funcionamento que é diferente do sistema real.

Com relação aos modelos gráficos probabilísticos na comunidade de robótica, (LU, 1997) foi o primeiro a introduzir o método dos mínimos quadrados para resolver a localização e o mapemamento simultâneos ou SLAM. Nesse importante artigo, os autores propuseram uma abordagem baseada em grafos no qual cada nó modela a posição e a orientação (pose) de um robô e as arestas representam as restrições entre as poses do robô. Estas restrições podem representar as medidas de odometria entre poses sequenciais ou no caso dos veículos aéreos, as medidas de uma unidade de medida inercial. Em (KUMMERLE, 2011) é apresentada a estrutura geral da otimização em grafos e uma implementação em C++. Contudo, trata-se de uma abordagem *offline*. Os primeiros a utilizar o modelo gráfico de grafos-fatores são Dellaert e Kaess em (DELLAERT, 2006).

Eles chamam seu método de \sqrt{SAM} (*Square Root SAM*) e utilizam métodos de fatoração de matrizes, tais como QR, LU e Cholesky para resolver o problema da otimização. Esta também é uma abordagem *offline*. Uma melhoria para o método \sqrt{SAM} anterior foi apresentada em (KAESS, 2008), onde novas medições são adicionadas usando atualização de matrizes via rotações de Givens (GOLUB, 2012). Contudo, apesar de ser uma abordagem *online*, atualizações *batch* periódicas são necessárias para permitir a reordenação das variáveis e a relinearização. Estas operações são custosas e prejudicam parte da solução *online*. Em (KAESS, 2012) foi demonstrado que calcular a estimativa MAP é equivalente à realização de inferência sobre o grafo-fator. Esse cálculo pode ser realizado de forma eficiente usando a técnica de suavização incremental. Essa abordagem explora a dispersão do sistema e a topologia do grafo-fator (KSCHISCHANG, 2001) otimizando apenas uma pequena parcela dos nós a cada atualização, diferentemente das abordagens de otimização *batch* ou suavização de janela fixa, que sempre recalculam todas as variáveis de estado. O método é uma nova visão sobre o trabalho (KAESS, 2008) onde as atualizações incrementais matriciais via rotações de Givens são substituídas pela inferência apenas na parcela afetada do grafo-fator quando uma nova medida é incorporada.

O artigo (DAPAI XIAO, 2011) apresenta a criação de uma plataforma básica de medidas de iniciais, sendo usada para o desenvolvimento de um piloto automático para os projetos de veículos aéreos não-tripulados do IME. A plataforma consiste não só do hardware que contém os sensores capazes de realizar as medições, mas também de um firmware capaz de estimar a atitude da plataforma, criando efetivamente um sistema AHRS (*Attitude Heading and Reference System*), bem como de um software com o qual é possível visualizar os dados gerados. Contudo, o trabalho foca na prototipação do hardware de medidas iniciais.

Em (GAO, 2013) foi abordado o uso de grafos-fatores para sistemas de navegação SINS/GPS/CNS (SINS - *Strapdown Inertial Navigation System* e CNS - *Constellation Navigation System*) apresentando simulações para os resultados encontrados. Este trabalho foca na navegação marítima.

Em (SUNDERHAUF, 2013, 2012a,b) foram apresentadas propostas para o uso de grafos-fatores para o problema dos múltiplos caminhos em sistemas GPS. Esse é um problema comum nesse contexto e foi comentado na subseção 3.2.6.

Já em (PENN, 2012), o principal objetivo foi o de fornecer uma análise do efeito da fusão de vários sensores em uma solução de navegação, abordando os conceitos e

objetivos do programa ASPN (*All Source Positioning and Navigation* (DARPA, 2010) visa a permitir soluções de navegação perfeitamente integradas, de baixo custo e robustas nos cenários militares, em qualquer plataforma operacional e em qualquer ambiente, com ou sem GPS, possibilitando uma rápida adaptação às missões em evolução, bem como reduzir os custos de integração dos sistemas de navegação em geral). Para isso foi utilizado o filtro de Kalman estendido considerando 6 cenários e 19 sensores. No entanto, como o próprio autor conclui, existem muitas desvantagens no uso do EKF. Uma delas foi que a implementação dos 19 sensores não foi trivial usando o EKF e a outra foi que a construção da matriz Jacobiana das medições acabou tornando-se um processo longo e tedioso. A vantagem da utilização dos grafos é que as informações dos sensores tornam-se nós adicionados ao grafo. Isso evita a construção tediosa de diversas combinações de matrizes, conforme o conjunto de sensores.

Com o trabalho de (INDELMAN, 2012) observamos, pela primeira vez, a formulação de grafos-fatores para o problema da navegação, permitindo o uso de sensores de diferentes taxas de operação e, possivelmente, assíncronos. O método proposto no artigo é demonstrado em um ambiente simulado usando uma unidade de medida inercial (*Inertial Measurement Unit* (IMU) - dispositivo onde são montados os sensores inerciais, que consistem em três acelerômetros e três girômetros. Os primeiros fornecem as medidas dos componentes da aceleração linear, enquanto que os últimos fornecem os componentes da velocidade angular), GPS e sensor de visão estéreo. Nesse caso não há uma aplicação prática e o conjunto de sensores é simplificado para que se possa testar o método.

Finalmente em (LANGE, 2013) encontramos a primeira aplicação prática da fusão de sensores via suavização incremental com grafos-fatores. O foco é na navegação *indoor* com o seguinte conjunto de sensores: IMU, fluxo óptico e sonar para as medições de altitude. Para a aplicação prática foi utilizado um quadricóptero medindo 72 cm de diâmetro e podendo transportar até 500 g de carga por cerca de 20 min. Nossa proposta visa a aplicação do método em um VANT de asa fixa e com um conjunto distinto de sensores.

3 TÓPICOS TUTORIAIS

Neste capítulo, apresentamos os conceitos utilizados como referencial teórico para o desenvolvimento deste trabalho. Os assuntos abordados envolvem tópicos relacionados à sistemas de referência, à sensores, à grafos, à modelos gráficos probabilísticos e à otimização.

3.1 SISTEMAS DE REFERÊNCIA

Um sistema de referência é um sistema de coordenadas no qual a posição, a orientação e outras propriedades dos objetos podem ser medidos e analisados. Para isso existem diversos desses sistemas, pois, muitas vezes, a definição de um vetor em um determinado sistema de referência faz mais sentido e é mais fácil de ser entendido do que em outro. O sistema de referência no qual um vetor é representado será indicado por um subscrito. Por exemplo, \mathbf{v}_i é o vetor da velocidade no sistema de referência i , ou inercial. Vetores e matrizes serão representados por caracteres em negrito. Descrevemos, nesta seção, os sistemas mais comuns no âmbito da navegação. Maiores detalhes podem ser obtidos em (FARRELL, 2008; GROVES, 2013).

O primeiro sistema a ser definido é conhecido como referencial ECI (*Earth-Centered Inertial*). Como podemos inferir pelo nome, este é um referencial inercial, onde as Leis do movimento de Newton podem ser aplicadas, ou seja, o referencial não está girando ou acelerando com respeito ao resto do Universo. Nesse sistema, a origem o_i é situada no centro da Terra e o eixo z_i sempre aponta ao longo do eixo de rotação da Terra, do centro para o norte verdadeiro (não magnético). Os eixos x_i e y_i situam-se no plano equatorial. O referencial inercial é importante em navegação porque os sensores iniciais medem o movimento em relação a esse referencial, possibilitando assim, a derivação das equações da navegação.

Outro sistema comum em navegação é o referencial ECEF (*Earth-Centered Earth-Fixed*). Este referencial também possui sua origem o_e no centro da Terra, porém os eixos rotacionam em relação ao referencial ECI (a velocidade angular da Terra é $\omega_e = 7,2921 \times 10^{-5} rad/s$). O eixo z_e aponta para o norte verdadeiro, o eixo x_e para a interseção entre o meridiano de Greenwich e a linha do Equador e o eixo y_e é perpendicular aos eixos

z_e e x_e .

Mais específico aos problemas aeronáuticos, o referencial NED (*North-East-Down*) é definido como um plano tangente à superfície da Terra que se move com a aeronave, porém seus eixos apontam para direções diferentes daquelas definidas pelo referencial do corpo da aeronave. Nesse sistema, o eixo x_n aponta para o norte verdadeiro, o eixo y_n aponta para o leste, enquanto que o eixo z_n aponta para o centro da Terra.

Por último, o referencial do corpo da aeronave é aquele que descreve o objeto que está navegando, possuindo origem o_b em seu centro de gravidade e coincidindo com o referencial NED. O eixo x_b aponta para a frente da aeronave, o eixo y_b aponta para o leste e o eixo z_b aponta para baixo da aeronave. A FIG. 3.1 ilustra os quatro referenciais citados: ECI, ECEF, NED e do corpo da aeronave.

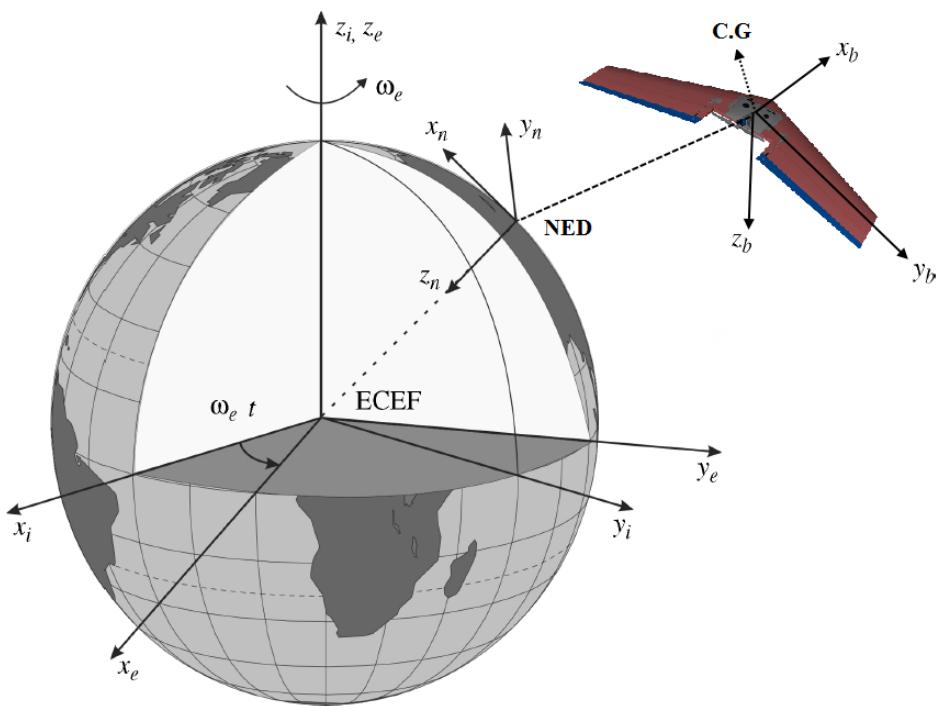


FIG. 3.1: Sistemas de referência ECI, ECEF, NED e do corpo. Adaptado de (FOSSEN, 2011)

Além destes, precisamos definir o referencial do simulador, pois o X-Plane[®] não adota os padrões aeronáuticos para o referencial do corpo da aeronave. Nesse caso, o referencial adotado é o EUS (*East-Up-South*), ou seja, o eixo x_s aponta para o leste da aeronave, o eixo y_s aponta para cima e o eixo z_s aponta para a parte de trás. O principal motivo para o uso desse sistema está no fato do simulador utilizar a biblioteca gráfica *OpenGL*

como seu principal meio de desenvolvimento. Ou seja, esse é o referencial adotado pela comunidade de visão computacional. Sua inclusão é importante, pois, não se pode utilizar nenhum dos dados provenientes do X-Plane® sem antes transformá-los para o referencial do corpo da aeronave. Este foi um dos primeiros problemas encontrados no início das simulações. A transformação é realizada com a seguinte operação:

$$\mathbf{x}_b = \mathbf{R}_s^b \mathbf{x}_s = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \mathbf{x}_s \quad (3.1)$$

onde \mathbf{x} é o vetor de estados da aeronave, discutido na próxima seção, e \mathbf{R}_s^b é a matriz de rotação que transforma o referencial de origem s para o referencial de destino b . A FIG. 3.2 ilustra os dois referenciais. O artigo (MCGHEE, 1988) trata justamente das relações

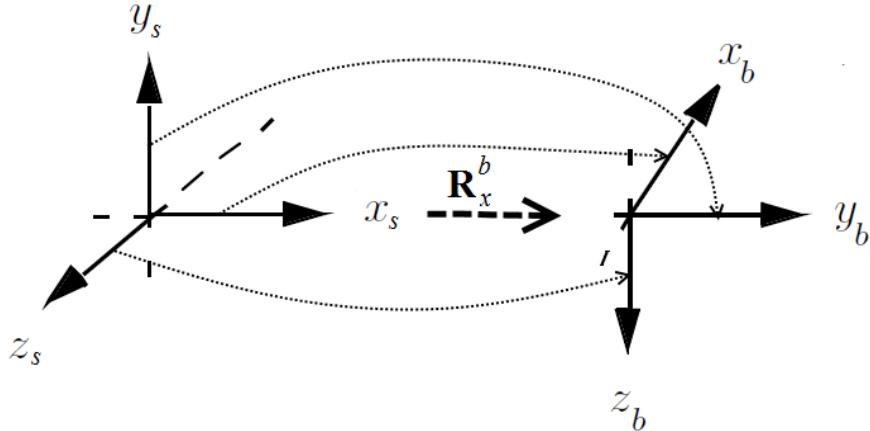


FIG. 3.2: Referencial do simulador em comparação com o referencial do corpo da aeronave.

entre as convenções gráficas e as aeronáuticas.

3.2 SENsores

Dado que precisamos estimar o estado da aeronave composto por nove variáveis - posição (p_x, p_y, p_z) , orientação (ϕ, θ, ψ) e velocidade (v_x, v_y, v_z) ¹ -, precisamos de estruturas especiais que possam colaborar ou medir de alguma forma essas informações. Essas estruturas são os sensores. Nesta seção, apresentamos o conjunto de sensores utilizados na navegação da aeronave.

¹O Capítulo 5 abordará as variáveis de estado da aeronave.

3.2.1 ACELERÔMETRO

Acelerômetros são dispositivos baseados na segunda Lei de Newton que medem forças específicas (consequentemente, acelerações, dado que $F = ma$). Logo, sua saída é dada por:

$$\mathbf{a}_m = \mathbf{a}_l + \mathbf{g} \quad (3.2)$$

onde \mathbf{a}_m é a aceleração medida, \mathbf{a}_l é a aceleração linear e \mathbf{g} é a aceleração da gravidade. Dessa forma, um acelerômetro em repouso ($\mathbf{a}_l = 0$) irá medir somente a gravidade na componente z . Portanto, a saída do acelerômetro, nesse caso, é $\mathbf{a}_m = \mathbf{g}$. Caso esteja em queda livre, a saída será $\mathbf{a}_m = 0$, pois $\mathbf{a}_l = -\mathbf{g}$. Normalmente, são encapsulados em um circuito integrado três acelerômetros dispostos ortogonalmente (um para cada eixo). Como estamos interessados na aceleração da aeronave somente, devemos remover os efeitos da gravidade

$$\mathbf{a}_l = \mathbf{a}_m - \mathbf{R}_i^b \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (3.3)$$

onde \mathbf{R}_i^b , é a matriz que transforma a gravidade do referencial inercial (nesse referencial a gravidade aponta para baixo ou o centro da Terra) para o referencial do corpo da aeronave.

3.2.2 GIRÔMETRO

Girômetros são dispositivos semelhantes aos giroscópios. No entanto, enquanto o primeiro mede a variação angular em seu próprio eixo ($\dot{\theta}_b(t)$), o último mede diretamente a sua orientação ($\theta(t)$) contra um ponto de referência fixo. Desse modo, ao usar um girômetro, precisamos integrar a velocidade angular para obter a orientação

$$\theta(t) = \int_0^t \dot{\theta}_b(t) dt + \theta_0 \quad (3.4)$$

onde θ_0 é a orientação inicial da aeronave. Contudo, como as medições da variação angular possuem ruídos, a integração faz com que os erros da orientação aumentem. Os dois sensores estão representados conceitualmente na FIG. 3.3.

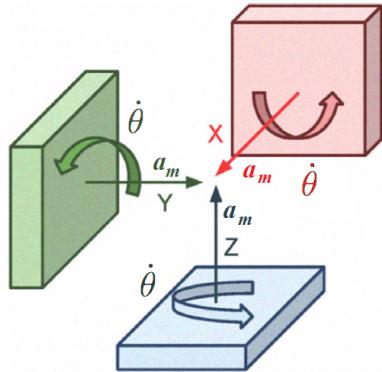


FIG. 3.3: Representação conceitual de um acelerômetro e um girômetro. Adaptado de (DAPAXÃO, 2011)

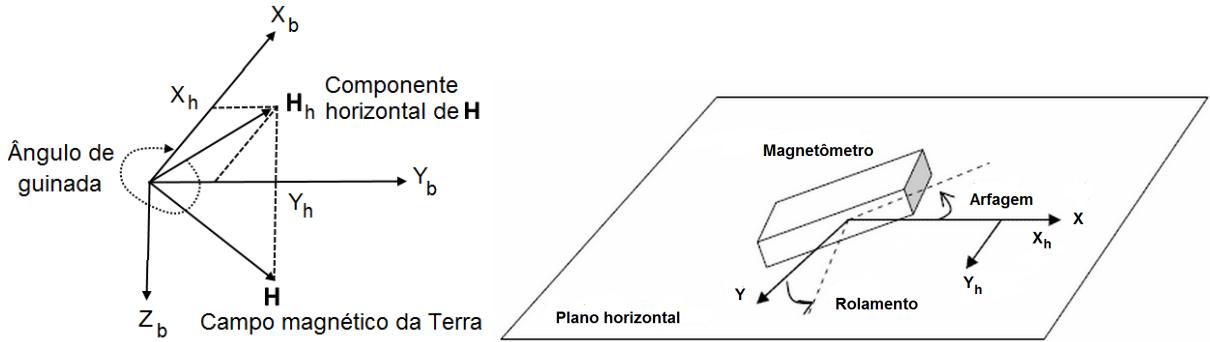
3.2.3 MAGNETÔMETRO

Magnetômetros são usados para medir o vetor do campo magnético da Terra (\mathbf{H}), sendo uma forma de se obter o ângulo de guinada² (ψ) da aeronave. Quando esta encontra-se na posição horizontal (ângulos de rolamento (ϕ) e arfagem (θ) iguais a zero), o ângulo de guinada magnético pode ser determinado pela equação

$$\psi_m = -\text{atan}2(Y_h, X_h) \quad (3.5)$$

onde X_h e Y_h são as medições realizadas pelo magnetômetro nos eixos x e y no plano horizontal. Neste plano, o vetor \mathbf{H}_h é a componente de \mathbf{H} que aponta para o norte magnético, conforme ilustrado na FIG. 3.4a. Contudo, como a aeronave muda constantemente de orientação, não podemos garantir o alinhamento dos sensores com o plano horizontal. Nesse caso, basta aplicar duas rotações para movê-lo para o plano horizontal, conforme a FIG. 3.4b. Matematicamente, temos

²O ângulo de guinada (ψ) é medido ao redor do eixo z, o ângulo de rolamento (ϕ), ao redor do eixo x e o ângulo de arfagem (θ), ao redor do eixo y. Maiores detalhes no Capítulo 5.



(a) Ângulo de guinada quando a aeronave encontra-se no plano horizontal.
(b) Movimentos realizados para trazer o magnetômetro para o plano horizontal

FIG. 3.4: Representação conceitual da medição do campo magnético.

$$\begin{aligned} \mathbf{H}_h &= \mathbf{R}_b^h(\phi, \theta) \mathbf{H} \\ &= \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix} \mathbf{H} \end{aligned} \quad (3.6)$$

$$\begin{pmatrix} X_h \\ Y_h \\ Z_h \end{pmatrix} = \begin{pmatrix} c_\theta & s_\theta s_\phi & s_\theta c_\phi \\ 0 & c_\phi & -s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix} \mathbf{H}$$

3.2.4 BARÔMETRO

Os barômetros medem a chamada altitude barométrica comparando a pressão atual com a pressão inicial, normalmente do ponto de decolagem da aeronave. Dessa forma, os barômetros respondem a mudanças na pressão atmosférica tomada a partir de um medidor de pressão estática. Assim, as medições da altura podem ser inferidas a partir de medições da pressão atmosférica, já que a pressão de um fluido (neste caso, o ar) pode ser definida como a medida da força por unidade de área exercida pelo fluido sobre uma superfície em contato. As medições da altura são consideravelmente importantes no contexto das aeronaves, uma vez que fornecem dados importantes ao pouso e também porque as informações de altitude provenientes de outros sensores como o GPS não são muito precisas. Mais especificamente, a variação na pressão estática, ΔP_s , é função da variação da altitude, Δh , da densidade do ar, ρ e da gravidade, g . Acontece que a densidade do

ar é função da pressão estática, P_s , e da temperatura do ar, T_{ar} , ou seja:

$$\frac{\Delta P_s}{\Delta h} = -g\rho = -g \frac{M}{RT_{ar}} P_s \quad (3.7)$$

onde M é a massa molecular do ar e R a constante universal dos gases. Para temperaturas constantes a EQ. 3.7 pode ser tratada como uma equação diferencial resultando numa solução para a pressão como função da altitude em relação a uma pressão inicial, P_0 (FIG. 3.5):

$$P_s = P_0 e^{-\frac{Mg}{RT_{ar}} \Delta h} \quad (3.8)$$

Isso quer dizer que se armazenarmos a pressão estática do ponto de partida do VANT, a

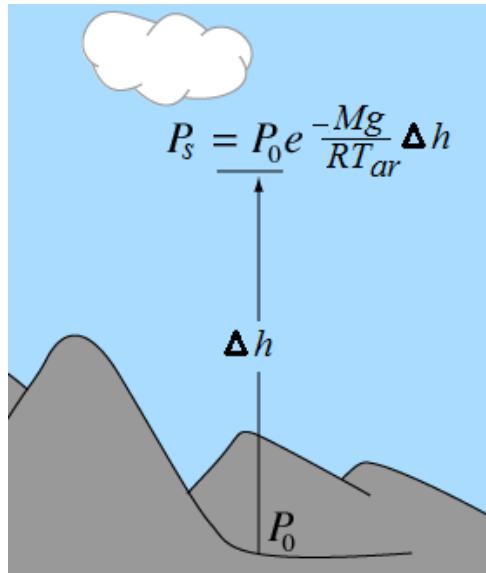


FIG. 3.5: Relação entre a variação da altura com a variação da pressão.

altura acima do ponto de partida, h_{ACP} estimada pelo barômetro é calculada pela equação

$$h_{ACP} = -\frac{RT_{ar}}{Mg} \ln(P_s/P_0) \quad (3.9)$$

3.2.5 SENSOR DE PRESSÃO DIFERENCIAL

Um outro sensor baseado em pressão é o sensor de pressão diferencial. Para isso utiliza-se um tubo de Pitot que fica em contato com a corrente de ar sendo usado para medir a velocidade relativa do ar. A pressão total medida através do tubo de pitot, P_{pitot} é a soma da pressão estática, P_s , com a pressão dinâmica, P_d . A Pressão dinâmica é proporcional à densidade do ar, ρ , e aumenta com o quadrado da velocidade do ar, V_{ar} :

$$P_d = P_{pitot} - P_s = \frac{1}{2} \rho V_{ar}^2 \quad (3.10)$$

Assim, a velocidade relativa do ar pode ser estimada rearranjando a EQ. 3.10

$$V_{ar} = \sqrt{2(P_{pitot} - P_s)/\rho} \quad (3.11)$$

A FIG. 3.6 mostra o funcionamento de um sensor de pressão diferencial com o auxílio do tubo de Pitot.

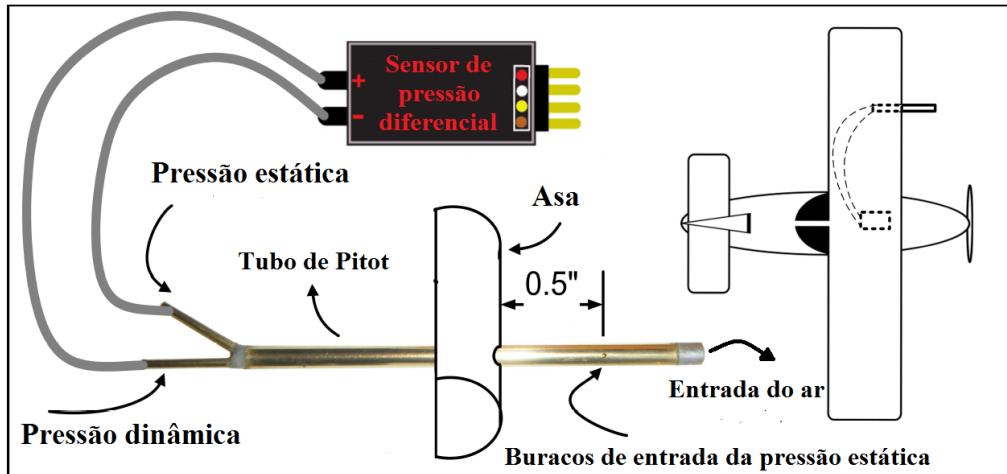


FIG. 3.6: Representação esquemática do funcionamento do sensor de pressão diferencial. Adaptado de (SYSTEMS, 2014).

3.2.6 GPS

O Sistema de Posicionamento Global (GPS)³ é um sistema de navegação baseado em satélites que fornecem informações sobre a posição 3D de objetos sobre ou perto da superfície da Terra. O principal componente do sistema de GPS é a constelação de 24 satélites que orbitam a Terra continuamente a uma altitude de aproximadamente 20.000 km. A configuração das órbitas de satélite é concebida de modo que qualquer ponto sobre a superfície da terra seja observável por, pelo menos, quatro satélites em todos os momentos. Por meio da medição dos tempos de percurso dos sinais a partir de um mínimo de quatro satélites para um receptor na proximidade da superfície da terra, a localização do receptor em três dimensões pode ser determinada. O tempo de percurso do sinal de onda de rádio é utilizado para determinar o intervalo de cada satélite para o receptor. Por que quatro medições? Porque com uma medição de um local conhecido, podemos localizar

³Além do sistema GPS americano, entrou em operação, no ano de 2011, o sistema russo denominado GLONASS. O sistema europeu Galileo e o sistema chinês BeiDou-2 encontram-se em desenvolvimento.

um ponto em uma linha (1D). Duas medições podem localizar um ponto em um plano (2D) e três medições podem localizar um ponto sobre uma superfície 3D. Veja a Figura 3.7.

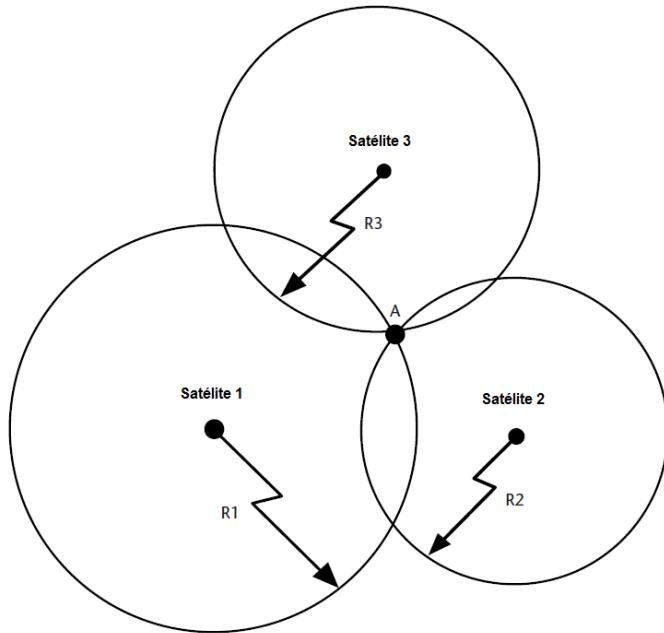


FIG. 3.7: Representação esquemática da determinação da posição GPS em 3D. Caso **sem** erros de sincronização satélite-receptor. (KAPLAN, 2005).

Acontece que devido a erros de sincronização entre os relógios dos satélites - que são atômicos e por isso quase perfeitamente sincronizados - e o relógio do receptor, precisamos de mais uma medição para triangular a posição do receptor. Assim, para resolver a posição em três dimensões com o erro de *offset* do relógio do receptor, são necessárias pelo menos quatro medições. A geometria associada às medidas de quatro satélites diferentes formam um sistema de quatro equações algébricas não lineares com quatro incógnitas: latitude, longitude e altitude do receptor GPS e o erro de *offset* do relógio do receptor. Esse situação está representada na figura 3.8.

Além do erro de sincronização dos relógios, ainda podemos listar os seguintes erros: atraso ionosférico (atrasos de propagação do sinal GPS causados pela ionosfera), atraso troposférico (atrasos ocorridos na propagação do sinal nas camadas mais baixas da atmosfera), erros de múltiplos caminhos (desvios e reflexões sofridos pelo sinal GPS até a recepção, tais como construções e árvores), erros de efemérides dos satélites (diferenças entre a posição real do satélite e aquela prevista por meio dos parâmetros orbitais fornecidos nas mensagens de navegação dos satélites) e erros de medição nos receptores (ocasiona-

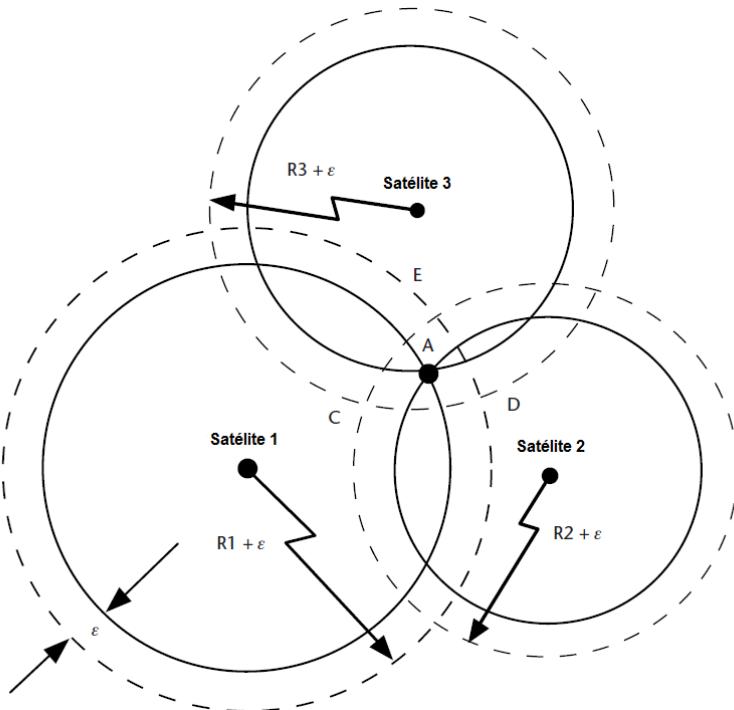


FIG. 3.8: Representação esquemática da determinação da posição GPS em 3D. Caso real com erros de sincronização satélite-receptor. (KAPLAN, 2005).

dos pelo hardware e pela antena). Um tratamento detalhado sobre o assunto pode ser encontrado em (GREWAL, 2007). Com efeito, devido aos diversos problemas do sistema GPS, a baixa taxa de operação (1 a 10 Hz), a vulnerabilidade à interferências (*jaming*), a sua inviabilidade em aplicações internas (*indoor*), seu uso *ad hoc* em sistemas de tempo real ou missão crítica é inviável. Por isso, a necessidade de um conjunto de sensores com a técnica da fusão de sensores.

3.3 GRAFOS: CONCEITOS BÁSICOS

É conveniente apresentar aqui algumas definições básicas da teoria de grafos. Um grafo é uma estrutura combinatória que consiste em um conjunto finito não vazio de objetos, chamados vértices, e um conjunto finito (possivelmente vazio) de pares ordenados de vértices, chamado de arestas ou arcos. Os grafos servem como uma maneira simples e compacta de representar estas relações em um computador, abrindo o caminho para a utilização de algoritmos computacionais eficientes. O material aqui apresentado foi baseado nas definições de (COWELL, 2006) e (VALIENTE, 2002).

Definição 1. Um grafo $G = (V, E)$ consiste em um conjunto finito não-vazio V de vértices

e um conjunto finito $E \subseteq V \times V$ de arestas ou arcos. Uma aresta $e = (v, w)$ é dita incidente aos vértices v e w , onde v é a origem e w o destino, e os vértices v e w são ditos **adjacentes** ou **vizinhos**.

Definição 2. Um grafo $G = (V, E)$ é dito **não-dirigido** se $(v, w) \in E$ implica em $(w, v) \in E$, para todo $v, w \in V$. Caso contrário, será um grafo **dirigido**.

Definição 3. Seja $G = (V, E)$ um grafo, e $W \subseteq V$. Um grafo (W, S) é um **subgrafo** de G se $S \subseteq E$. O subgrafo de G **induzido** por W é o grafo $(W, E \cap W \times W)$.

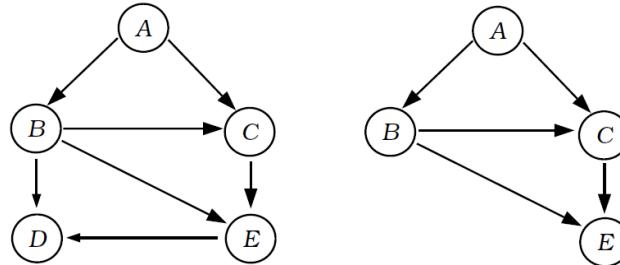


FIG. 3.9: Um grafo dirigido G e o subgrafo induzido por W .

Definição 4. Um grafo não-dirigido $G = (V, E)$ é dito **bipartido** se V pode ser partitionado em dois subconjuntos U e W , de tal forma que para todo $(v, w) \in E$, ou $v \in U$ e $w \in W$, ou $v \in W$ e $w \in U$.

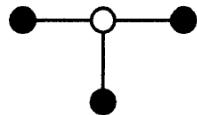


FIG. 3.10: Um grafo bipartido. Os vértices V estão partitionados nos subconjuntos $U = \{\text{vértices pretos}\}$ e $W = \{\text{vértices brancos}\}$

Definição 5. Um grafo não-dirigido $G = (V, E)$ é dito **completo** se para todo $v, w \in V$ com $v \neq w$, $(v, w) \in E$.

Definição 6. Um grafo não-dirigido $G = (V, E)$ é **conectado** se para todo par de vértices $v, w \in E$, existe um caminho entre v e w .

Definição 7. Um grafo conectado $G = (V, E)$ é dito uma **árvore** se o grafo não-dirigido subjacente não possui ciclos e há um nó distinto $r \in V$, chamado de raiz da árvore, de tal forma que para todos os nós $v \in V$, existe um caminho em G do nó raiz r até o nó v .

Definição 8. Seja $T = (V, E)$ uma árvore. O nó $v \in V$ é dito o **pai** do nó $w \in V$, denotado por $pa(w)$, se $(v, w) \in E$ e, em tal caso, o nó w é dito ser um nó **filho** de v .

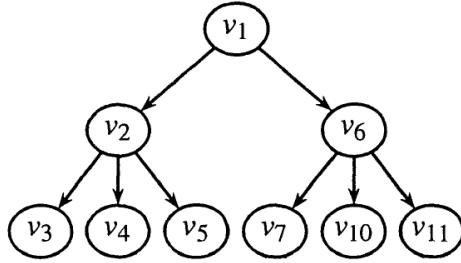


FIG. 3.11: Uma árvore com o nó raiz v_1 . O nó v_2 é filho de v_1 e pai dos nós v_3, v_4 e v_5 .

Definição 9. Um **clique** de um grafo não-dirigido $G = (V, E)$ é um conjunto de vértices $C \subseteq V$ de tal forma que o subgrafo de G induzido por C é completo, ou seja, tal que $\{v, w\} \in E$ para todos os vértices distintos $v, w \in C$.

3.4 MODELOS GRÁFICOS PROBABILÍSTICOS

Ao longo das últimas décadas, os modelos gráficos probabilísticos tornaram-se um dos principais métodos para a representação da incerteza. Diversas áreas de pesquisa como visão computacional, processamento de voz, ciência cognitiva, processamento de sinais e robótica passaram a tirar proveito desses modelos. Modelos gráficos probabilísticos (PGM) são representações de distribuições de probabilidade sobre um conjunto de variáveis. Eles usam a representação da teoria dos grafos, onde os nós correspondem às variáveis aleatórias e as arestas expressam as relações probabilísticas entre as variáveis. Dessa forma, os PGM fornecem uma estrutura poderosa para modelar a distribuição de probabilidade conjunta de um grande número de variáveis aleatórias. Nesta seção fornecemos uma breve introdução a dois modelos gráficos probabilísticos: as redes Bayesianas e os grafos-fatores.

3.4.1 REDES BAYESIANAS

Redes Bayesianas (PEARL, 1998) são grafos acíclicos dirigidos que representam variáveis aleatórias e suas dependências condicionais. Para cada dependência condicional entre duas variáveis, existe uma aresta dirigida que liga os dois nós correspondentes. Dessa forma, a semântica das redes Bayesianas especificam a distribuição de probabilidade conjunta dada pelo produto

$$P(x_1, \dots, x_n) = P(X) = \prod_i P(x_i | pa_i) \quad (3.12)$$

onde X forma o conjunto de todas as variáveis aleatórias e pa_i , as variáveis cujos nós pais estão diretamente ligados ao nó x_i . A figura 3.12 ilustra uma rede Bayesiana que representa a função de probabilidade conjunta

$$P(x_1, x_2, x_3, x_4, x_5, x_6) = P(x_1|x_2)P(x_2|x_3)P(x_3|x_4)P(x_4)P(x_5|x_1, x_2)P(x_6|x_1, x_3, x_4) \quad (3.13)$$

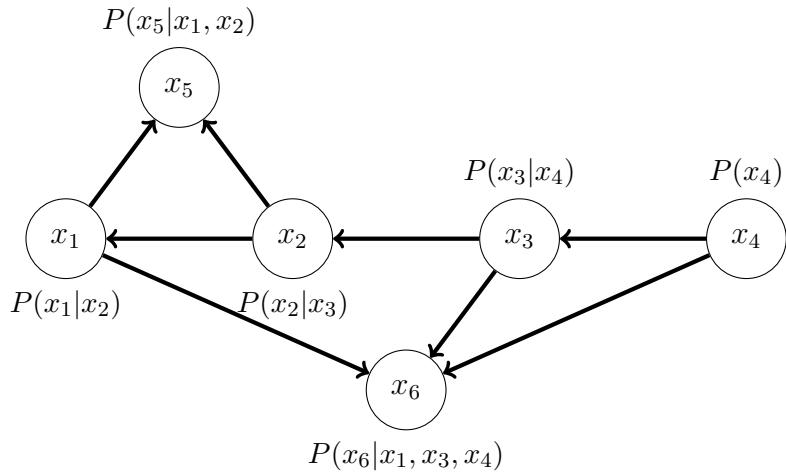


FIG. 3.12: Rede bayesiana representando a função de probabilidade conjunta $P(x_1, x_2, x_3, x_4, x_5, x_6) = P(x_1|x_2)P(x_2|x_3)P(x_3|x_4)P(x_4)P(x_5|x_1, x_2)P(x_6|x_1, x_3, x_4)$.

3.4.2 GRAFOS-FATORES

Um grafo-fator (KSCHISCHANG, 2001) $G = (\mathcal{F}, \mathcal{X}, \mathcal{E})$ é um grafo bipartido com dois tipos de nós: nós de fatores $f_i \in \mathcal{F}$ e nós de variáveis $x_j \in \mathcal{X}$. Arestas $e_{ij} \in \mathcal{E}$ somente existem entre os nós de fatores e os nós de variáveis, e são presentes se e somente se o fator f_i envolve uma variável x_j . Assim, o grafo-fator G define uma fatoração da função

$f(\mathcal{X})$ como:

$$f(\mathcal{X}) = \prod_i f_i(\mathcal{X}_i), \quad (3.14)$$

onde \mathcal{X}_i é o conjunto de todas as variáveis x_j conectadas por uma aresta a um fator f_i . Como exemplo, considere a função $f(x_1, x_2, x_3) = f_1(x_1)f_2(x_1, x_2)f_3(x_1, x_3)$. Seu respectivo grafo-fator é mostrado na figura 3.13.

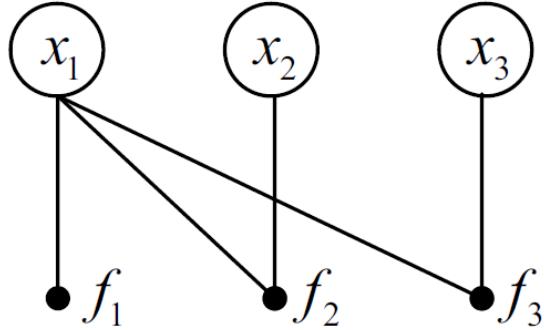


FIG. 3.13: Grafo-fator para a função $f(x_1, x_2, x_3)$.

Maiores detalhes sobre modelos gráficos probabilísticos podem ser encontrados em (KOLLER, 2009; BISHOP, 2006). Para aplicações no campo da robótica, uma boa referência é (THRUN, 2005).

3.5 OTIMIZAÇÃO VIA MÍNIMOS QUADRADOS

Nesta seção apresentamos o problema geral da otimização e uma de suas mais importantes classes: a otimização de funções representadas por uma soma de termos quadráticos. Esse problema é conhecido como uma otimização via mínimos quadrados. Nossa objetivo não é prover discussões detalhadas sobre os diversos aspectos, métodos e técnicas de otimização, mas sim, uma introdução necessária para a resolução de alguns problemas da robótica. Para uma fonte de consulta rápida recomendamos (MADSEN, 2004) e para um maior aprofundamento destacamos o livro (NOCEDAL, 2006).

Um problema de otimização é definido como encontrar um mínimo de uma função custo ou função objetivo $F(\mathbf{x})$. Dessa forma, não estamos interessados no valor de $F(\mathbf{x})$ no mínimo, mas sim, qual o valor da variável \mathbf{x}^* onde esse mínimo ocorre. Formalmente, podemos definir um problema de otimização como

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} F(\mathbf{x}) \quad (3.15)$$

onde $\mathbf{x} \in \mathbb{R}^n$ é um vetor de números reais com $n \geq 1$ e $F : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função diferenciável. Na otimização via mínimos quadrados, a função objetivo é representada por uma soma de termos quadráticos

$$F(\mathbf{x}) = \frac{1}{2} \sum_i^n f_i(\mathbf{x})^2 \quad (3.16)$$

onde $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. De maneira mais compacta e conveniente, vetorialmente temos

$$F(\mathbf{x}) = \frac{1}{2} \mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) \quad (3.17)$$

onde $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))^\top$. Dessa forma a otimização passa a ser

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} F(\mathbf{x}) = \arg \min_{\mathbf{x}} \frac{1}{2} \mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) \quad (3.18)$$

Essa classe de otimização é de grande importância para a resolução de diversos problemas, dentre eles, os da robótica. No Capítulo 4, onde trataremos da formulação de nosso problema, iremos ver que a navegação de um robô, ou mais especificamente, de um VANT, pode ser inserida nesse contexto. Nos casos mais simples, a função objetivo $\mathbf{f}(\mathbf{x})$ é linear e consideramos que todas as informações possuem o mesmo peso. Assumem assim a seguinte forma:

$$f_i(\mathbf{x}) = a_{i,1}x_1 + a_{i,2}x_2 \dots a_{i,n}x_n - b_i \quad (3.19)$$

Ou, se agruparmos todos os coeficientes $a_{i,n}$ na matriz \mathbf{A} e todas as constantes b_i no vetor \mathbf{b} , vetorialmente teremos

$$\mathbf{f}(\mathbf{x}) = \mathbf{Ax} - \mathbf{b} \quad (3.20)$$

Na prática, nos problemas da robótica, a função objetivo é não linear e queremos levar em consideração o grau de confiança de cada informação. São os chamados problemas não lineares ponderados. Os detalhes para a resolução desses problemas de otimização encontram-se no Apêndice A. De maneira resumida, o mínimo de uma função pode ser obtido calculando-se a derivada primeira e igualando-a a zero. No caso dos problemas não lineares, antes de calcularmos a derivada primeira, devemos linearizar a função em torno de um ponto mediante o emprego da expansão de Taylor de primeira ordem. Dessa forma, os sistemas a serem resolvidos são:

- a) Problemas lineares: $\mathbf{A}^\top \mathbf{Ax}^* = \mathbf{A}^\top \mathbf{b}$
- b) Problemas lineares ponderados: $\mathbf{A}^\top \Omega \mathbf{Ax}^* = \mathbf{A}^\top \Omega \mathbf{b}$

- c) Problemas não lineares: $\mathbf{J}^\top \mathbf{J} \Delta \mathbf{x} = -\mathbf{J}^\top \mathbf{f}(\mathbf{x})$
- d) Problemas não lineares ponderados: $\mathbf{J}^\top \Omega \mathbf{J} \Delta \mathbf{x} = -\mathbf{J}^\top \Omega^\top \mathbf{f}(\mathbf{x})$

onde a diferença está na presença da matriz de informação Ω (definida como o inverso da matriz de covariâncias Σ); e na substituição da matriz \mathbf{A} pela matriz Jacobiana \mathbf{J} , no caso dos problemas não lineares. A matriz Jacobiana resulta do processo de linearização da matriz \mathbf{A} . A resolução dos quatro problemas acima pode ser obtida com o uso das decomposições matriciais como a decomposição QR e a decomposição *Cholesky*.

Após a introdução dos conceitos de otimização e modelos gráficos probabilísticos, estamos em boa posição para formularmos nosso problema, assunto do próximo capítulo.

4 FORMULAÇÃO DO PROBLEMA

O problema consiste em determinar uma estimativa para as variáveis de estado $\mathbf{X} = \{p_n, p_e, p_d, v_n, v_e, v_d, \phi, \theta, \psi\}$ do VANT, dado um conjunto de observações ou medições \mathbf{Z} realizadas por diversos sensores, considerando também, os comandos \mathbf{U} ⁴ sobre os atuadores ou controladores, que determinam ações sobre o VANT, portanto, afetando as variáveis de estado \mathbf{X} . Probabilisticamente, para a trajetória completa $\mathbf{X}_{1:t}$, essa estimativa pode ser formulada como

$$P(\mathbf{X}_{1:t} | \mathbf{U}_{1:t}, \mathbf{Z}_{1:t}) \quad (4.1)$$

sendo esse problema conhecido como suavização (*smoothing*). Ao contrário, se desejarmos apenas o estado atual considerando apenas as medições no instante desse estado, o problema passa a ser o cálculo da distribuição de probabilidade

$$P(\mathbf{X}_t | \mathbf{U}_t, \mathbf{Z}_t) \quad (4.2)$$

sendo essa abordagem conhecida como filtragem. Imediatamente percebemos que possuímos um compromisso entre a qualidade e a performance. Nesse sentido, as abordagens baseadas em filtragem dominaram durante um bom tempo as aplicações robóticas onde a performance era crucial. No entanto, dados os avanços no processamento computacional e algoritmos cada vez mais otimizados, passou-se a considerar o emprego da suavização. As técnicas atuais de otimização via suavização representam o estado da arte em sistemas robóticos, sendo foco de constantes pesquisas. Neste capítulo iremos ver como a fusão dos sensores do VANT pode ser vista como uma otimização via método dos mínimos quadrados (visto genericamente no Capítulo 11.1). Em seguida, no próximo capítulo, veremos como obter performance, mesmo considerando todas as informações disponíveis.

⁴Os comandos \mathbf{U} representam a odometria da aeronave que em nosso caso é função das medidas inerciais (forças específicas e velocidades angulares).

Dessa forma, irei considerar apenas o cálculo da EQ. 4.1, que pode ser reescrita como

$$\begin{aligned} P(\mathbf{X}_{1:t} | \mathbf{U}_{1:t}, \mathbf{Z}_{1:t}) &= \prod_i P(x_i | u_i, z_i) \\ &= \underbrace{\prod_i P(x_i | x_{i-1}, u_i)}_{\text{modelo do movimento}} \cdot \underbrace{\prod_k P(z_k | x_k)}_{\text{modelo das medições}} \end{aligned} \quad (4.3)$$

onde o índice $(1:t)$ será omitido para denotarmos o cálculo sobre todo o conjunto. Nossa probabilidade agora é o produto entre dois produtórios. Isso quer dizer que nosso problema pode ser fatorado em dois tipos diferentes de restrições. A primeira são as restrições de movimento que associam dois estados consecutivos x_i e x_{i-1} via um modelo tipicamente não linear do movimento $f(\cdot)$,

$$x_i \sim \mathcal{N}(f(x_{i-1}, u_i), \Sigma_i) \quad (4.4)$$

e a outra restrição associa a medição realizada com a posição atual via um modelo de medição $h(\cdot)$, também geralmente não linear.

$$z_k \sim \mathcal{N}(h(x_k), \Lambda_k) \quad (4.5)$$

Como é comum nos problemas robóticos e de estimação, os modelos seguem uma distribuição Normal ou Gaussiana, onde Σ_i e Λ_k são as matrizes de covariância associadas ao modelo do movimento e da medição, respectivamente.

4.1 FORMULAÇÃO VIA MÉTODO DOS MÍNIMOS QUADRADOS

Como vimos, o nosso grafo de navegação representa a probabilidade $P(\mathbf{X}_{1:t} | \mathbf{U}_{1:t}, \mathbf{Z}_{1:t})$, que será abreviada para $P(\mathbf{X} | \mathbf{U}, \mathbf{Z})$. Dado o conjunto de medições e o movimento realizado pelo VANT, buscamos a configuração ótima, isto é, a configuração mais provável dos estados do VANT, que denotamos por \mathbf{X}^* . Essa configuração mais provável é chamada de *máximo a posteriori* ou estimativa MAP. É igual à moda da distribuição de probabilidade $P(\mathbf{X} | \mathbf{U}, \mathbf{Z})$. Em outras palavras, \mathbf{X}^* é o ponto em que essa distribuição tem o seu máximo:

$$\mathbf{X}^* = \arg \max_X P(\mathbf{X} | \mathbf{U}, \mathbf{Z}) \quad (4.6)$$

$$\begin{aligned} &= \arg \max_X \underbrace{\prod_i P(x_i | x_{i-1}, u_i)}_{\text{modelo do movimento}} \cdot \underbrace{\prod_k P(z_k | x_k)}_{\text{modelo das medições}} \end{aligned} \quad (4.7)$$

De acordo com a EQ. 4.4 e a EQ. 4.5, os modelos do movimento e das medições são Gaussianos, ou seja,

$$x_i \sim \mathcal{N}(f(x_{i-1}, u_i), \Sigma_i)$$

$$P(x_i|x_{i-1}, u_i) = \frac{1}{\sqrt{2\pi|\Sigma_i|}} \exp\left(-\frac{1}{2}(f(x_{i-1}, u_i) - x_i)^T \Sigma_i^{-1} (f(x_{i-1}, u_i) - x_i)\right)$$

O termo $\frac{1}{\sqrt{2\pi|\Sigma_i|}}$ é constante e pode ser substituído por η , um normalizador. Utilizando, ainda, a definição da distância de Mahalanobis⁵ obtemos a simplificação

$$P(x_i|x_{i-1}, u_i) = \eta \cdot \exp\left\|-\frac{1}{2}(f(x_{i-1}, u_i) - x_i)\right\|_{\Sigma_i}^2 \quad (4.8)$$

Adotando o mesmo procedimento para o modelo das medições obtemos

$$P(z_k|x_k) = \eta \cdot \exp\left\|-\frac{1}{2}(h(x_k) - z_k)\right\|_{\Lambda_k}^2 \quad (4.9)$$

Com isso nossa probabilidade pode ser escrita como

$$P(\mathbf{X}|\mathbf{U}, \mathbf{Z}) \propto \prod_i \exp\left\|-\frac{1}{2}(f(x_{i-1}, u_i) - x_i)\right\|_{\Sigma_i}^2 + \prod_k \exp\left\|-\frac{1}{2}(h(x_k) - z_k)\right\|_{\Lambda_k}^2 \quad (4.10)$$

Aplicando o logaritmo negativo, transformamos a equação numa soma

$$-\log P(\mathbf{X}|\mathbf{U}, \mathbf{Z}) \propto \sum_i \|(f(x_{i-1}, u_i) - x_i)\|_{\Sigma_i}^2 + \sum_k \|(h(x_k) - z_k)\|_{\Lambda_k}^2 \quad (4.11)$$

As funções não lineares dentro da norma ou distância de Mahalanobis, são as funções que representam o erro dos movimentos e o erro das medições. Com essa observação podemos simplificar a EQ. 4.11 para

$$-\log P(\mathbf{X}|\mathbf{U}, \mathbf{Z}) \propto \sum_i \|\mathbf{e}(x_{i-1}, u_i, x_i)\|_{\Sigma_i}^2 + \sum_k \|\mathbf{e}(x_k, z_k)\|_{\Lambda_k}^2 \quad (4.12)$$

$$\propto \sum_i \|\mathbf{e}_i^f\|_{\Sigma_i}^2 + \sum_k \|\mathbf{e}_k^h\|_{\Lambda_k}^2 \quad (4.13)$$

onde \mathbf{e}_i^f e \mathbf{e}_k^h , representam os erros associados aos movimentos e às medições, respectivamente.

⁵Lembremos que a distância de Mahalanobis é definida como $\|a - b\|_{\Sigma}^2 = (a - b)^T \Sigma^{-1} (a - b)$.

Resumindo os procedimento descritos anteriormente nosso problema pode ser escrito como:

$$\mathbf{X}^* = \arg \max_X P(\mathbf{X} | \mathbf{U}, \mathbf{Z}) \quad (4.14)$$

$$= \arg \min_X -\log P(\mathbf{X} | \mathbf{U}, \mathbf{Z}) \quad (4.15)$$

$$= \arg \min_X \sum_i \| (f(x_{i-1}, u_i) - x_i) \|_{\Sigma_i}^2 + \sum_k \| (h(x_k) - z_k) \|_{\Lambda_k}^2 \quad (4.16)$$

$$= \arg \min_X \sum_i \| \mathbf{e}_i^f \|_{\Sigma_i}^2 + \sum_k \| \mathbf{e}_k^h \|_{\Lambda_k}^2 \quad (4.17)$$

O resultado acima é uma otimização via mínimos quadrados de uma função de erro não linear, uma vez que temos uma minimização sobre uma soma de termos ao quadrado, como queríamos demonstrar. Além disso, vimos que a EQ. 4.1 podia ser fatorada nas probabilidades associadas aos modelos do movimento e das medições conforme a EQ. 4.3. Essa fatoração nos permite visualizar a navegação do VANT como um grafo-fator. Esse modelo gráfico foi discutido na Seção 3.4.2. A FIG. 4.1 ilustra um grafo-fator simplificado para nosso problema, onde os fatores que representam as medições estão em verde e os fatores que representam o movimento estão em azul.

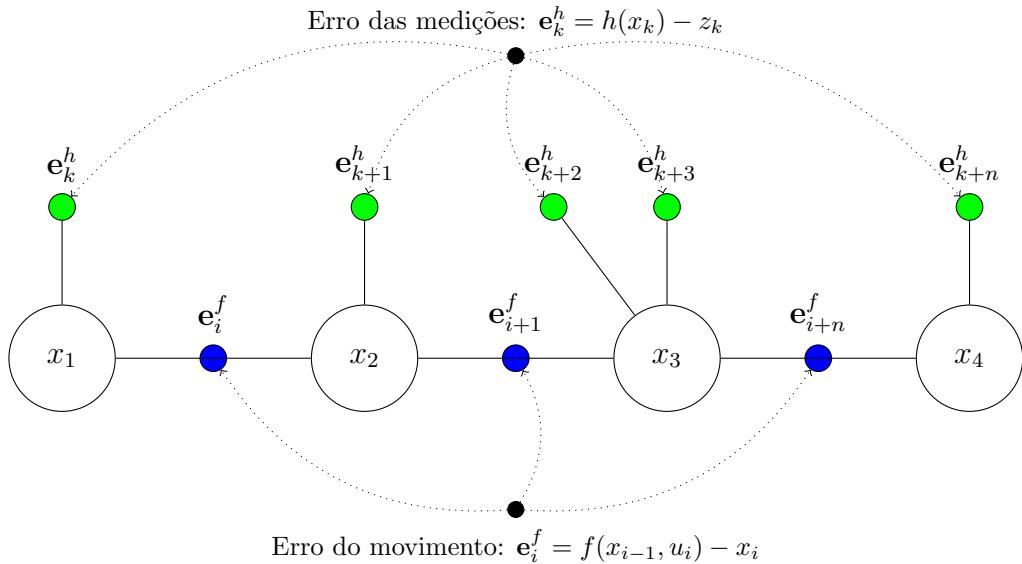


FIG. 4.1: Grafo-fator simplificado ilustrando os fatores que associam duas posições consecutivas (modelo do movimento, em azul) e os fatores que associam um estado com as medições (modelo das medições, em verde).

Da exposição contida no Capítulo 11.1, se coletarmos todos os erros do movimento e das medições no vetor \mathbf{e} , as matrizes de covariância na matriz de informação Ω e os

Jacobianos na matriz \mathbf{J} obtemos

$$\mathbf{J}^\top \Omega \mathbf{J} \Delta \mathbf{x} = -\mathbf{J}^\top \Omega \mathbf{e} \quad (4.18)$$

$$\mathbf{H} \Delta \mathbf{x} = -\mathbf{b} \quad (4.19)$$

onde $\mathbf{H} = \mathbf{J}^\top \Omega \mathbf{J}$ e $\mathbf{b} = \mathbf{J}^\top \Omega$. Outra forma de reescrevermos a EQ. 4.17 é feita mediante a eliminação das matrizes de covariância na norma de Mahalanobis utilizando

$$\|\mathbf{e}\|_\Sigma^2 = \mathbf{e}^T \Sigma^{-1} \mathbf{e} = \mathbf{e}^T \Sigma^{-1/2} \Sigma^{-1/2} \mathbf{e} = \|\Sigma^{-1/2} \mathbf{e}\|_2^2 \quad (4.20)$$

ou no caso dos erros das medições

$$\|\mathbf{e}\|_\Lambda^2 = \|\Lambda^{-1/2} \mathbf{e}\|_2^2 \quad (4.21)$$

Isso equivale a pré-multiplicarmos as funções de erro pela raiz quadrada das matrizes de covariância. Se pensarmos no caso escalar, isso significa dividir os erros pelos seus respectivos desvios padrão. Assim a EQ. 4.18 passa a ser escrita como

$$\Sigma^{1/2} \mathbf{J} \Delta \mathbf{x} = -\Sigma^{1/2} \mathbf{e} \quad (4.22)$$

$$\mathbf{A} \Delta \mathbf{x} = -\mathbf{a} \quad (4.23)$$

onde $\mathbf{A} = \Sigma^{1/2} \mathbf{J}$ e $\mathbf{a} = \Sigma^{1/2} \mathbf{e}$. Essa forma também é um sistema linear que pode ser obtido diretamente pelas decomposições Cholesky ou QR. Essa derivação é conhecida como *Square root smoothing and mapping* (\sqrt{SAM}), uma vez que utilizamos a raiz quadrada para substituirmos a norma de Mahalanobis pela norma l^2 . Essa derivação é importante, pois, é a usada pelo algoritmo de suavização incremental. Os detalhes podem ser encontrados em (DELLAERT, 2006). Finalmente, para compreendermos melhor, veremos um exemplo simplificado.

Exemplo 4.1. Considere uma aeronave deslocando-se a 10m/s. O modelo do movimento (odometria) possui $\sigma^{odo} = 1m$ e taxa de 1Hz. Inicialmente ela se encontra na posição $x_0 = 0$. Esse conhecimento prévio possui $\sigma^{prior} = 0,1m$. Além disso, a aeronave conta com um GPS que opera na frequência de 0,5Hz, ou seja, recebemos medições a cada 2 segundos. O GPS possui $\sigma^{gps} = 0,5m$. Com essas informações podemos construir o grafo-fator da FIG. 4.2 que representa o deslocamento da aeronave no intervalo de 0 a 4s. A TAB. 4.1 apresenta as medições provenientes da odometria e dos sensores.

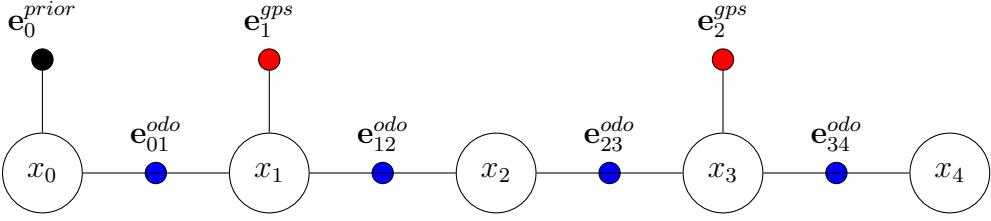


FIG. 4.2: Grafo-fator para o exemplo. Ele representa o movimento de uma aeronave no plano \mathbb{R}^1 .

Os métodos iterativos necessitam de uma estimativa inicial ou um ponto de partida para que a busca pela melhor solução seja feita. Para facilitar, utilizaremos como estimativas iniciais $x_0 = x_1 = x_2 = x_3 = x_4 = 0$. Na prática, nas soluções de navegação, informamos as condições iniciais como a posição de partida e a velocidade inicial. A partir daí, os próximos pontos de iteração usam a estimativa do passo anterior.

Medição	Valor
u_{01}^{odo}	10,0414
u_{12}^{odo}	9,2658
u_{23}^{odo}	9,9692
u_{34}^{odo}	10,2323
z_0^{prior}	0,0436
z_1^{gps}	9,8136
z_2^{gps}	29,8818

TAB. 4.1: Medidas provenientes da odometria e dos sensores.

Esse problema pode ser resolvido com a EQ. 11.21

$$\begin{aligned} \mathbf{J}^\top \Omega \mathbf{J} \Delta \mathbf{x} &= -\mathbf{J}^\top \Omega^\top \mathbf{f}(\mathbf{x}) \\ \mathbf{H} \Delta x &= -\mathbf{b} \end{aligned}$$

Utilizaremos a forma \sqrt{SAM} quando vermos a parte da suavização incremental. Para a

solução precisamos calcular os erros associados e seus respectivos Jacobianos:

$$\mathbf{f}(\mathbf{x}) = \mathbf{e}(\mathbf{x}) = \begin{bmatrix} \mathbf{e}_{01}^{odo} = -x_0 + x_1 - u_{01}^{odo} \\ \mathbf{e}_{12}^{odo} = -x_1 + x_2 - u_{12}^{odo} \\ \mathbf{e}_{23}^{odo} = -x_2 + x_3 - u_{23}^{odo} \\ \mathbf{e}_{34}^{odo} = -x_3 + x_4 - u_{34}^{odo} \\ \mathbf{e}_0^{prior} = x_0 - z_0^{prior} \\ \mathbf{e}_1^{gps} = x_1 - z_1^{gps} \\ \mathbf{e}_2^{gps} = x_3 - z_2^{gps} \end{bmatrix} = \begin{bmatrix} -10,0414 \\ -9,2658 \\ -9,9692 \\ -10,2323 \\ -0,0430 \\ -9,8136 \\ -29,8818 \end{bmatrix}$$

$$\mathbf{J} = \begin{array}{c|ccccc} & x_0 & x_1 & x_2 & x_3 & x_4 \\ \mathbf{e}_{01}^{odo} & -1 & 1 & 0 & 0 & 0 \\ \mathbf{e}_{12}^{odo} & 0 & -1 & 1 & 0 & 0 \\ \mathbf{e}_{23}^{odo} & 0 & 0 & -1 & 1 & 0 \\ \mathbf{e}_{34}^{odo} & 0 & 0 & 0 & -1 & 1 \\ \mathbf{e}_0^{prior} & 1 & 0 & 0 & 0 & 0 \\ \mathbf{e}_1^{gps} & 0 & 1 & 0 & 0 & 0 \\ \mathbf{e}_2^{gps} & 0 & 0 & 0 & 1 & 0 \end{array}$$

Em seguida, construimos a matriz de informação:

$$\begin{aligned} \Omega &= \begin{bmatrix} (\sigma_{01}^{odo})^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & (\sigma_{12}^{odo})^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & (\sigma_{23}^{odo})^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & (\sigma_{34}^{odo})^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & (\sigma_0^{prior})^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & (\sigma_1^{gps})^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & (\sigma_2^{gps})^2 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix} \end{aligned}$$

Realizando os cálculos obtemos:

$$\mathbf{H} = \begin{bmatrix} 101 & -1 & 0 & 0 & 0 \\ -1 & 6 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 6 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 5,7414 \\ -40,0300 \\ 0,7034 \\ -119,2641 \\ -10,2323 \end{bmatrix} \quad (4.24)$$

Resolvendo o sistema por decomposição QR, a solução obtida é:

$$\mathbf{x}^* = \mathbf{R}^{-1} \mathbf{Q}^\top \mathbf{b} = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 & x_4 \\ 0,0415 & 9,9311 & 19,5150 & 29,8023 & 40,0346 \end{bmatrix}^\top \quad (4.25)$$

O resultado é próximo da trajetória real:

$$\mathbf{x} = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 & x_4 \\ 0 & 10 & 20 & 30 & 40 \end{bmatrix}^\top \quad (4.26)$$

A FIG. 4.3 ilustra as funções de densidade de probabilidade das posições. Ela demonstra o grau de certeza de cada estimativa. Quanto mais achatada a curva mais precisa a informação. Assim percebemos que no início temos uma boa certeza da localização da aeronave em função do conhecimento prévio. Depois vemos que a cada movimentação a aeronave perde informações e que a cada sensoriamento do GPS ganhamos informação.

Em resumo, vimos que podemos representar o movimento da aeronave por um grafo, transformá-lo num sistema de mínimos quadrados e resolvê-lo por decomposições QR ou Cholesky. O que acontece quando recebemos uma nova informação? Devemos recalcular todo o grafo novamente. Claramente, trata-se de uma solução inviável para sistemas de tempo real. Para resolver o problema, ao invés de recalcular todo o grafo, a solução incremental, somente processa as partes afetadas por novas informações. Para entendermos o que isso representa, a FIG. 4.4 mostra o tempo de processamento em função da quantidade de fatores do grafo para as soluções batch e incremental. É possível verificar que a solução incremental não depende do tamanho do grafo. Ao contrário, a solução batch cresce linearmente. Os dados são provenientes do experimento com voo quadrado que serão discutidos no Capítulo 8. Os detalhes desse procedimento serão dados no próximo capítulo. A TAB. 4.2 apresenta a média e o máximo dos tempos de processamento para as soluções batch e incremental.

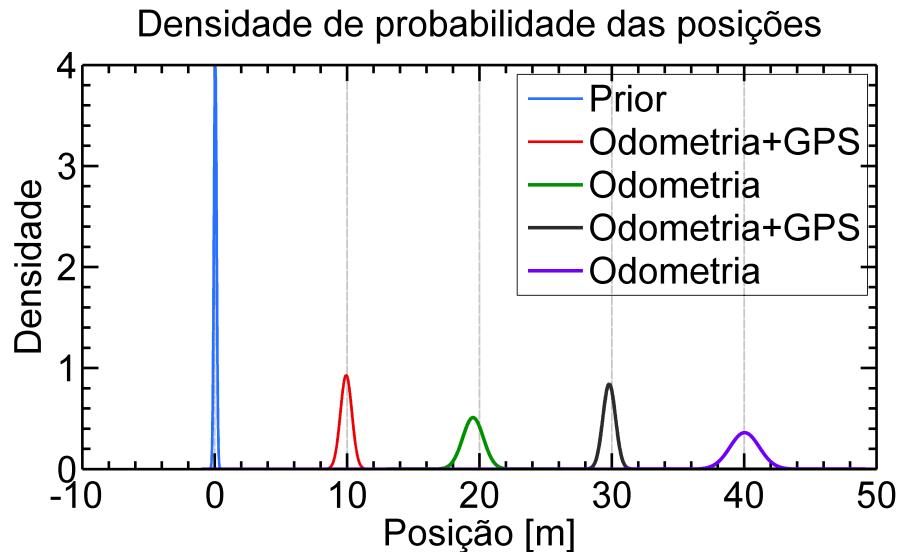


FIG. 4.3: Representação das funções de densidade de probabilidade das posições da aeronave.

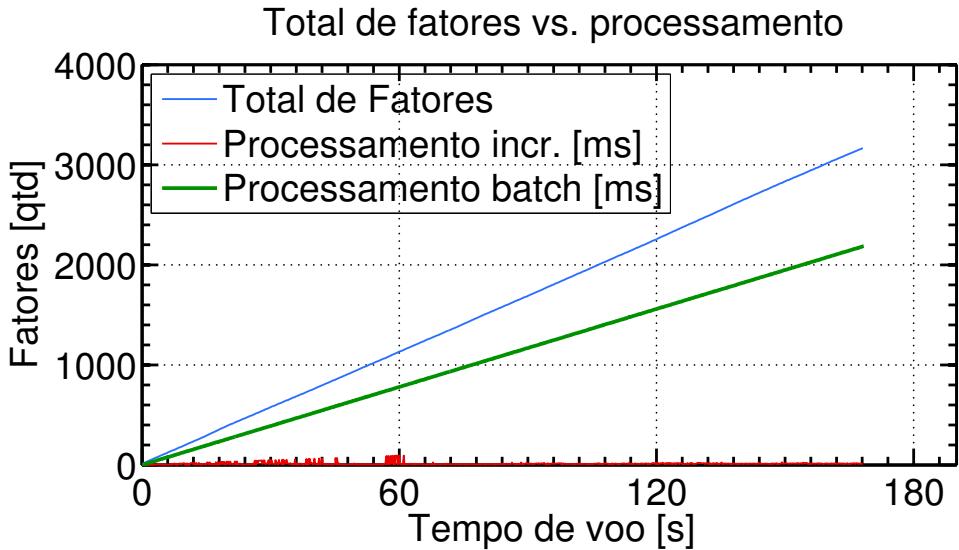


FIG. 4.4: Tempo de processamento em função da quantidade de fatores. O processamento incremental possui complexidade de tempo constante enquanto que a solução batch cresce linearmente com a quantidade de fatores.

Processamento	Tempo médio [ms]	Tempo máximo [ms]	Média de fatores recalculados [qtd]	Máx. de fatores recalculados [qtd]
Incremental	12,5	94,5	113,5044	1150
Batch	964,6	2289,2	1589,5	3167

TAB. 4.2: Tempos de processamento para as soluções batch e incremental.

5 VEÍCULO AÉREO NÃO TRIPULADO

Segundo (PASTOR, 2007), um VANT é composto de seis submódulos: a estrutura aerodinâmica do VANT; um sistema computacional embarcado; o controlador de missão; a carga útil; a estação de controle em solo e o sistema de comunicação. Neste capítulo descrevemos os quatro primeiros submódulos. Os dois últimos serão descritos no Capítulo 7. Nosso trabalho é justamente auxiliar na execução de um voo autônomo atuando no sistema computacional embarcado para realizar uma de suas tarefas mais importantes: a fusão dos sensores de maneira a obter as estimativas da posição, orientação e velocidade da aeronave.

5.1 ESTRUTURA AERODINÂMICA

A modelagem aerodinâmica é a principal e mais complexa tarefa a ser realizada na construção de um VANT. Por esse mesmo motivo, nossa intenção não foi modelar um novo VANT, assunto que foge do escopo desta dissertação. Para conhecer os desafios inerentes à modelagem aerodinâmica, o livro de (COOK, 2012) é uma boa referência. Ao contrário, procuramos um modelo aerodinâmico já consolidado no mercado nacional e de fácil obtenção. Dessa forma, nossa tarefa ficaria mais concentrada na fusão dos sensores. Para isso, deveríamos escolher um modelo de aeronave que fosse de fácil obtenção no mercado nacional e ao mesmo tempo estivesse disponível nos melhores simuladores de voo. O simulador de voo escolhido foi o X-Plane®. A escolha deveu-se a três condições: a possibilidade de escrever plugins; a possibilidade de modelar e importar novas aeronaves e a existência de trabalhos anteriores do IME que adotam esse simulador, em especial o trabalho de (ANGONESE, 2013b). Os detalhes sobre o simulador de voo X-Plane® serão abordados na Seção 7.2.1. Isso nos fez optar por uma abordagem reversa. Primeiro escolheríamos o modelo simulado para em seguida adquirir a asa. Essa decisão foi fundamental, pois evitou que adentrassemos tanto na modelagem aerodinâmica do simulador quanto da aeronave. Se escolhessemos uma aeronave que não possuísse um modelo aerodinâmico disponível no simulador, precisaríamos fazê-lo. Ao contrário, se escolhessemos um modelo simulado que não estivesse disponível no mercado nacional, precisaríamos importar a asa, ou na pior das hipóteses, construí-la do zero. Como esse não era o foco da pesquisa,

optamos pelo modelo de asa fixa com formato delta. Esse modelo é bastante comum nas lojas de aeromodelos e barato. Por esse mesmo motivo, também não foi difícil encontrar a aerodinâmica para o X-Plane®. O modelo simulado foi adaptado do trabalho de (NAUDIN, 2014). A FIG. 5.1 apresenta o modelo de asa fixa em formato delta, suas dimensões e seu centro de gravidade.

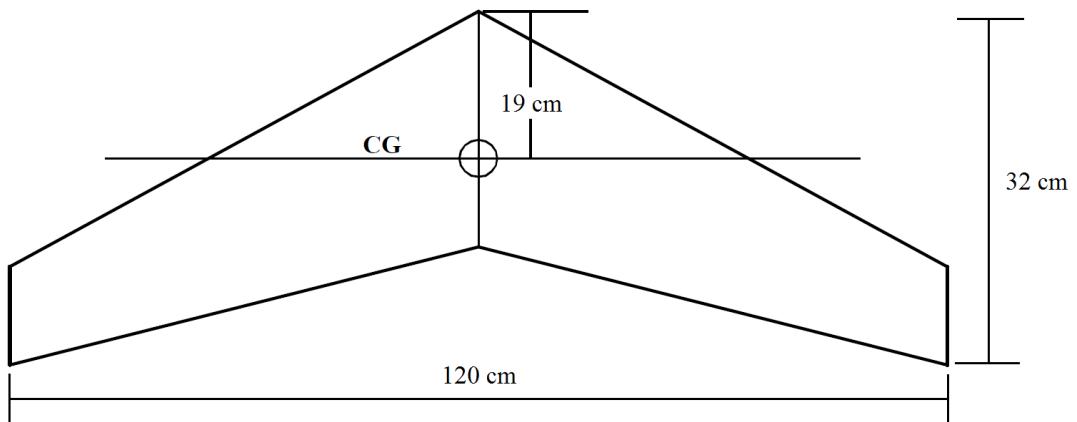


FIG. 5.1: VANT de asa fixa em formato delta, suas dimensões e seu centro de gravidade.

5.2 MOVIMENTOS E ESTADOS DA AERONAVE

Uma aeronave pode executar seis tipos de movimento em relação aos três eixos de referência, possuindo dessa forma seis graus de liberdade, sendo três movimentos lineares e três movimentos de rotação. Os movimentos lineares ou de translação são:

- a) para frente e para trás ao longo do eixo de rolamento
- b) para a esquerda e para a direita ao longo do eixo de arfagem
- c) para cima e para baixo ao longo do eixo de guinada

Os movimentos de rotação são:

- a) Ao redor do eixo de rolamento
- b) Ao redor do eixo de arfagem
- c) Ao redor do eixo de guinada

Com esses movimentos, podemos visualizar doze variáveis de estado⁶. Há três estados de posição e três estados de velocidade associados ao movimento de translação. Da mesma forma, existem três ângulos e três estados de velocidade angular associados ao movimento de rotação. As doze variáveis de estado e os movimentos da aeronave estão listadas na TAB. 5.1 e mostradas esquematicamente na FIG. 5.3. As posições do VANT (p_n, p_e, p_d) são definidas em relação ao referencial inercial. As velocidades lineares (u, v, w), as velocidades angulares (p, q, r) e os ângulos de Euler (rolamento ϕ , arfagem θ e guinada ψ) são definidos em relação ao referencial do corpo fixo do VANT.

Nome	Descrição
p_n	Posição inercial Norte (<i>North</i>)
p_e	Posição inercial Leste (<i>East</i>)
p_d	Posição inercial Para Baixo (<i>Down</i>)
u	Velocidade no referencial do corpo fixo medida ao longo do eixo \mathbf{i}^b
v	Velocidade no referencial do corpo fixo medida ao longo do eixo \mathbf{j}^b
w	Velocidade no referencial do corpo fixo medida ao longo do eixo \mathbf{k}^b
ϕ	Ângulo de rolamento
θ	Ângulo de arfagem
ψ	Ângulo de guinada
p	Velocidade de rolamento medida ao longo do eixo \mathbf{i}^b
q	Velocidade de arfagem medida ao longo do eixo \mathbf{j}^b
r	Velocidade de guinada medida ao longo do eixo \mathbf{k}^b

TAB. 5.1: Variáveis de estado necessárias para a localização do VANT.

Assim, para que os movimentos possam ser realizados, a aeronave conta com estruturas especiais denominadas superfícies de controle: os ailerons, os profundores e o leme. Os ailerons são estruturas móveis localizadas nas extremidades das asas responsáveis pelo movimento de rolamento. Os profundores são responsáveis pelo movimento de arfagem. O leme está localizado ao final da cauda da aeronave sendo responsável pelo movimento de guinada. A FIG. 5.2 apresenta essas estruturas.

No caso do modelo de asa no formato delta é utilizada uma superfície de controle resultante da combinação entre o aileron e o profundor (*elevator*). Por isso essa estrutura de controle recebeu o nome de elevon (*elevator + aileron*), sendo comum em aeronaves sem cauda, como no caso das de asa delta.

⁶Variáveis de estado são as variáveis que desejamos estimar e que contém as informações necessárias sobre a situação da aeronave.

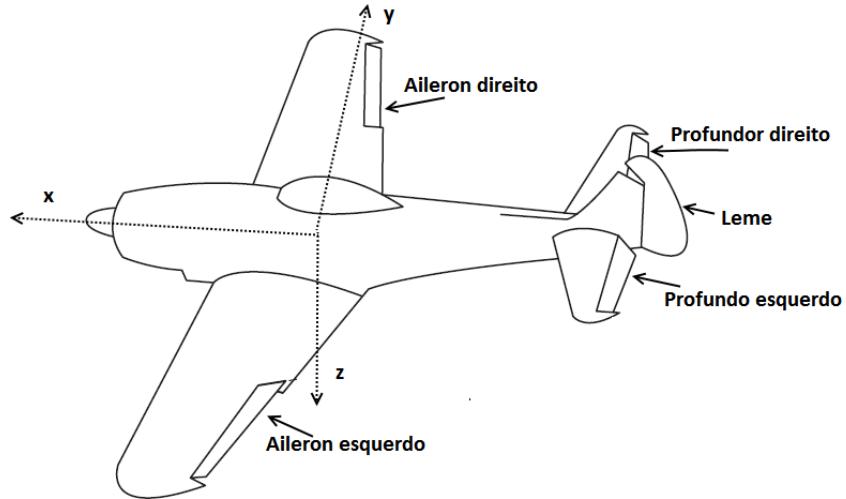


FIG. 5.2: Superfícies de controle de uma aeronave. Adaptado de (COOK, 2012)

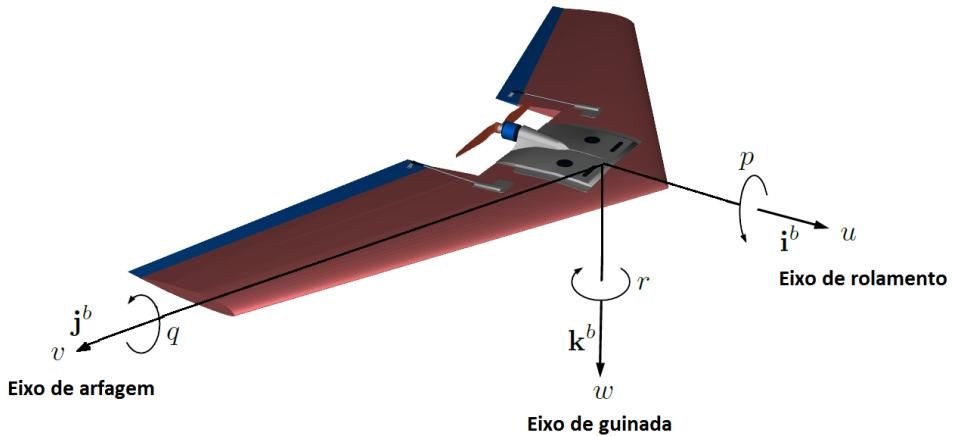


FIG. 5.3: Representação esquemática dos eixos de rotação e translação do VANT.

Os elevons simplificam a construção da aeronave uma vez que são necessários somente o controle de dois elevons. Esse controle é realizado por servomotores, dispositivos eletromecânicos que apresentam movimento proporcional a um comando. A operação de todas as superfícies de controle necessitaria de cinco servomotores, ao passo que os elevons necessitam de apenas dois. Dessa forma, reduz-se o peso e o consumo para a operação da aeronave. A desvantagem é que perdemos a possibilidade de execução de movimentos mais finos ou precisos. Logo, é necessária a combinação de movimentos dos elevons para deslocar a aeronave. A FIG. 5.4 mostra essas combinações para a realização das manobras de subida, descida, virada para a esquerda e virada para a direita.



(a) Movimento de subida. Os dois elevons apontam para cima. (b) Movimento de descida. Os dois elevons apontam para baixo.



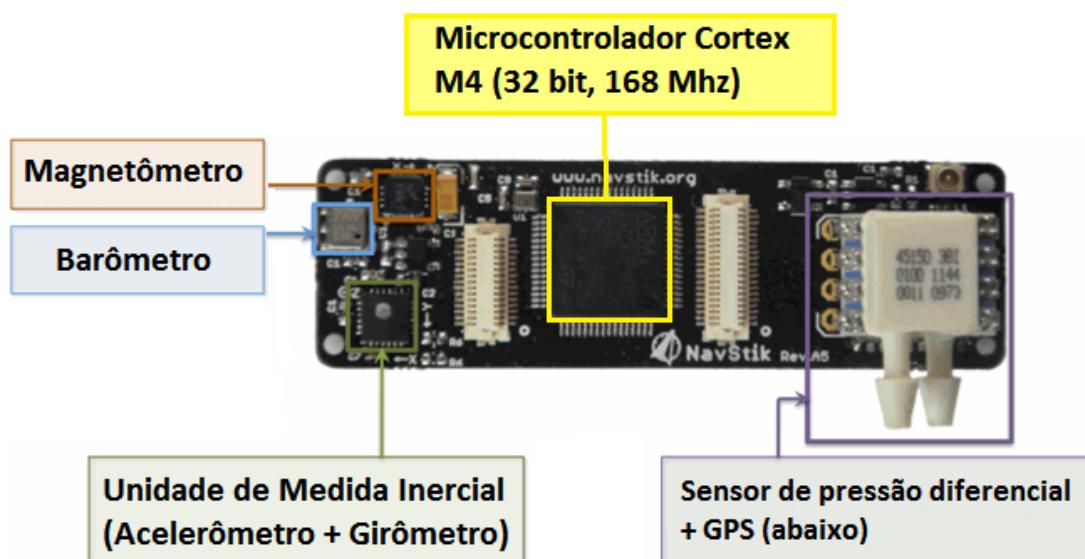
(c) Movimento para a esquerda. O elevon da esquerda aponta para cima e o da direita para baixo. (d) Movimento para a direita. O elevon da direita aponta para cima e o da esquerda para baixo.

FIG. 5.4: Movimentos realizados pelo VANT em função dos elevons.

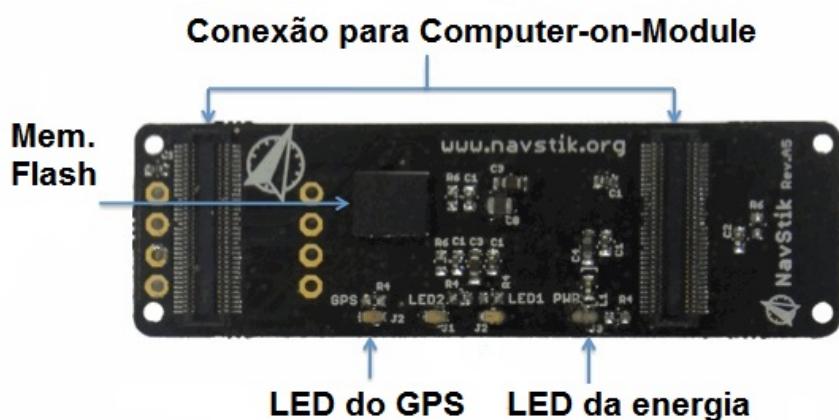
5.3 PILOTO AUTOMÁTICO NAVSTIK

Navstik (NAVSTIK, 2014) é um placa desenvolvida para a navegação e o controle de sistemas robóticos (FIG. 5.5). Para isso ela conta com uma ampla gama de sensores de navegação (acelerômetro, giroscópio, magnetômetro, sensores de pressão atmosférica, sensor de pressão diferencial e GPS) e um microcontrolador (32-bit Cortex M4). Tudo isso embutido em um placa de dimensões e peso bastante reduzidos (59 mm x 18 mm; $\sim 5,4$ g). Além disso, existe a capacidade de expansão possibilitando a integração com outros sensores e, principalmente, com módulos de processamento externo conhecidos

como *Computer-On-Module* (COM), caso haja necessidade de um poder computacional maior, caso da nossa solução de navegação ou outras como as que envolvem visão computacional. Além das importantes características citadas, outro fator crucial que fez com que escolhessemos o módulo Navstik foi a possibilidade de aplicação em um microVANT: aeronaves que possuem envergadura de no máximo 20 cm e peso total de 200g. Este é um projeto que vem sendo conduzido pelo Laboratório de Robótica do IME (FIG. 5.6). O software que roda o piloto automático é chamado de PandaPilot (PANDAPILOT, 2014). Ele é uma adaptação do projeto de código aberto PIXHAWK (MEIER, 2011) de modo a executar e trabalhar perfeitamente com os sensores do Navstik.



(a) Navstik frente.



(b) Navstik verso.

FIG. 5.5: Navstik com seu microcontrolador cortex M4 e os sensores.

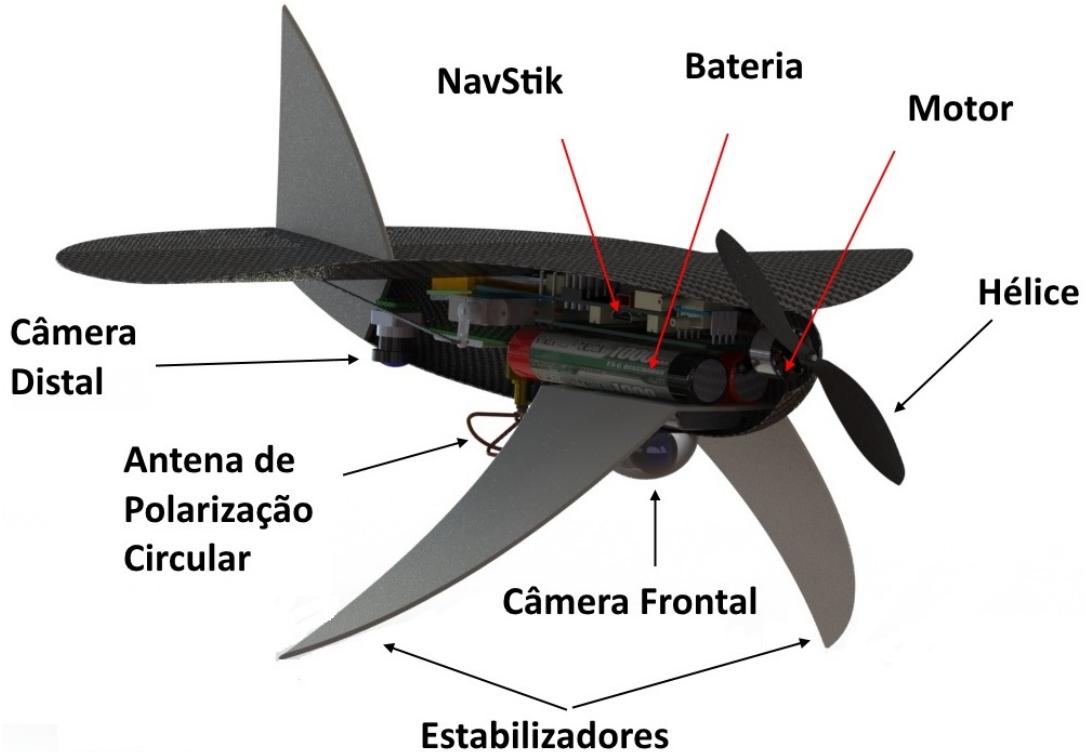


FIG. 5.6: Visão artística interna do projeto microVANT do laboratório de Robótica do IME. Fonte: Alberto Angonese e Victor Betio.

Para que o módulo Navstik possa ser operacionalizado é necessário uma placa de expansão. Essa placa, denominada IvyPRO pelo fabricante, proporciona quase todos os recursos necessários para o desenvolvimento de um VANT autônomo ou qualquer outro sistema robótico. Uma de suas características avançadas é o suporte à placa *Computer-On-Module* Gumstix Overo® que aumenta o poder computacional do Navstik. Dentre as funcionalidades da placa IvyPro destacamos:

- 6 canais PWM para comunicação com rádio controle.
- Suporte para cartão MicroSD para gravação de logs.
- 1 porta para telemetria (UART).
- 2 portas USB para atualização de firmware e para energia (durante o desenvolvimento).
- 2 portas USB e uma porta UART para comunicação com a placa Gumstix Overo®.
- 1 porta para operações de debug.

- Entrada de tensão na faixa de 4,5V - 14,0V.
- Peso de aproximadamente 8,6g.
- Tamanho de 41mm x 35 mm.

Com o uso das duas placas em conjunto, o peso total passa a ser de aproximadamente 14g, o que é bastante leve dada as suas potencialidades.

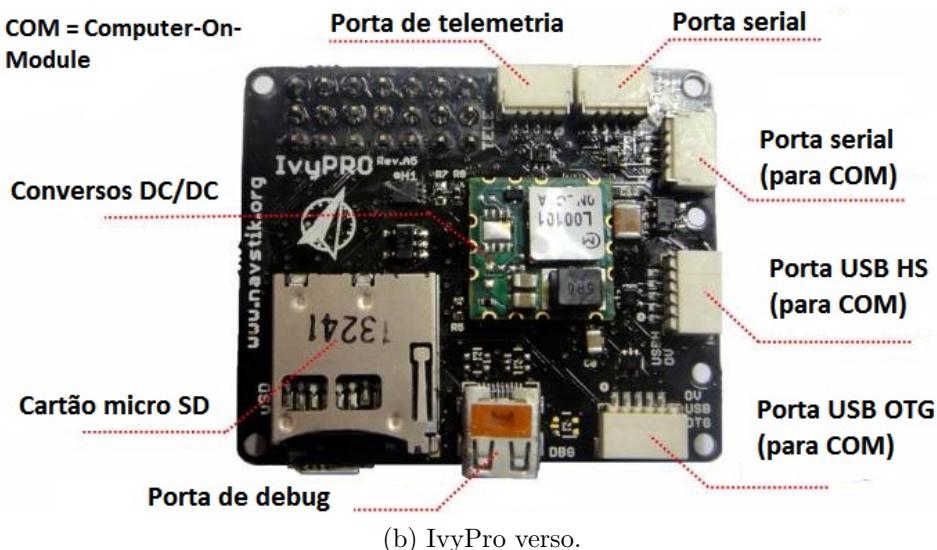
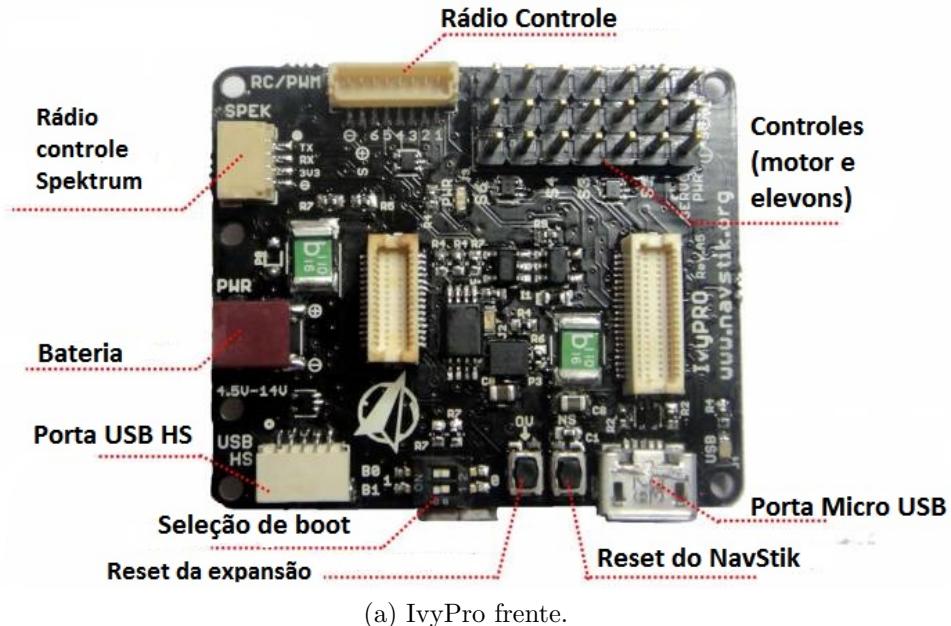


FIG. 5.7: Placa de expansão IvyPro. Responsável por prover as entradas e saídas para a placa Navstik.

5.4 VISÃO ESQUEMÁTICA

A aeronave completa conta com os seguintes componentes perfazendo um total de 289.8g (não inclui o peso da aeronave):

- (1) Piloto automático NavStik $\sim 5,4\text{g}$;
- (2) Placa de expansão IvyPro $\sim 8,6\text{g}$;
- (3) Bateria LiPo Zippy Compact de 2200mAh $\sim 163\text{g}$;
- (4) Dois servos Hextronik HX5010 para controle dos elevons $\sim 5,4\text{g}$ cada;
- (5) Receptor do rádio HK6S 6 canais 2.4Ghz $\sim 6,5\text{g}$
- (6) Módulo de telemetria XBee Pro 900Mhz $\sim 5,0\text{g}$
- (7) Motor Brushless Emax CF2812 $\sim 39\text{g}$
- (8) Hobbywing FlyFun 18A Brushless ESC $\sim 19\text{g}$
- (9) Hélice de 7 polegadas $\sim 5\text{g}$
- (10) Antena GPS patch ativa $\sim 6,0\text{g}$
- (11) Antena Xbee $\sim 3,5\text{g}$
- (12) Antena CloverLeaf 5.8 Ghz $\sim 4,0\text{g}$
- (13) Boscam FPV 5.8G Video Module RX5808 $\sim 10\text{g}$
- (14) Mini câmera 782x572 11mm $\sim 1\text{g}$
- (15) Tubo de Pitot Prandtl $\sim 3\text{g}$

Para melhor visualizar a interação desses componentes elaboramos o modelo esquemático da FIG. 5.8.

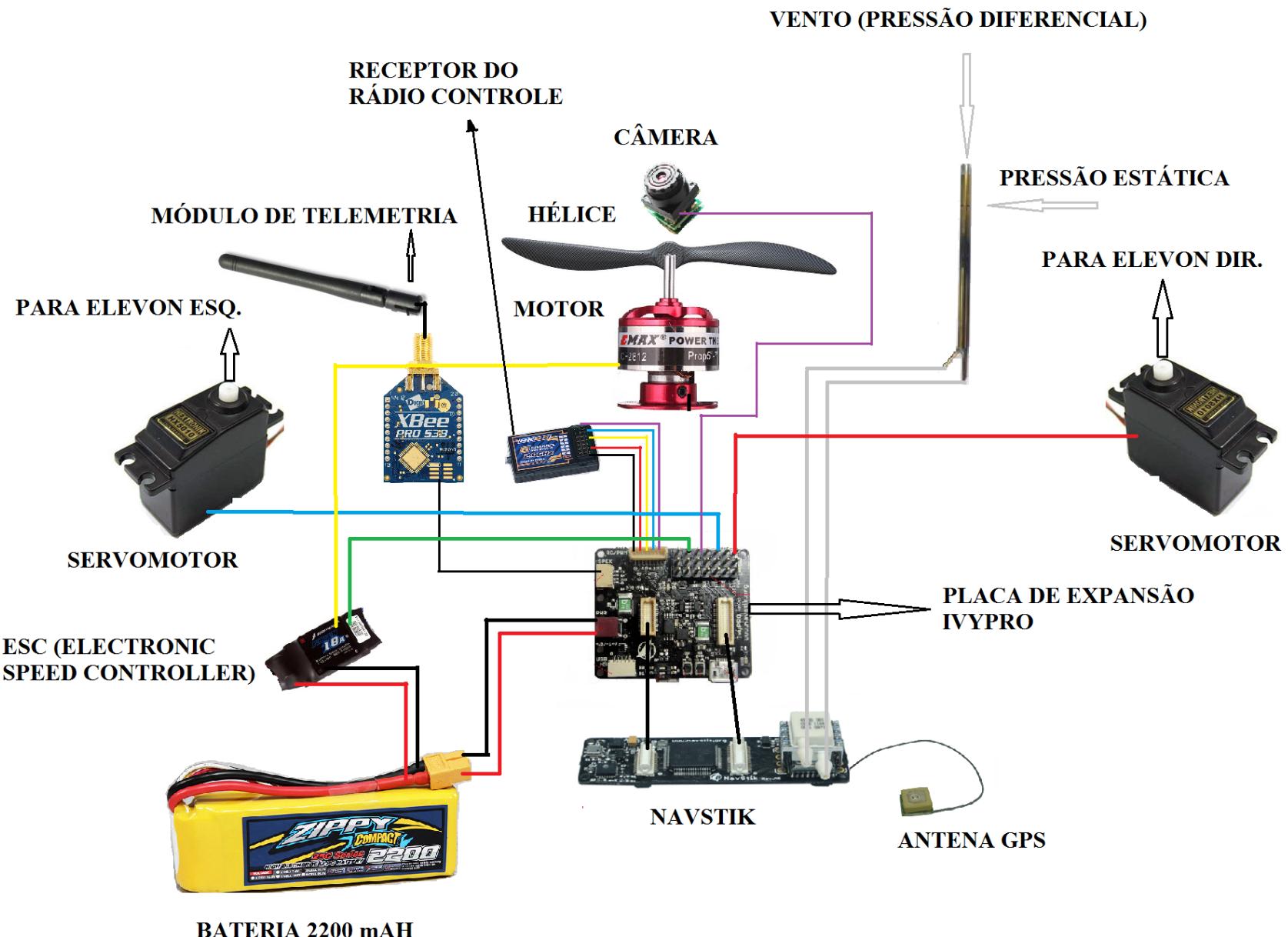


FIG. 5.8: Modelo esquemático dos componentes do VANT.

6 SOLUÇÃO: PRÉ-INTEGRAÇÃO E SUAVIZAÇÃO INCREMENTAL

Neste capítulo iremos ver como obter performance no cálculo do grafo de navegação. Duas soluções são apresentadas: a suavização incremental, que atua no back-end e a pré-integração, que atua no front-end. A primeira solução evita o cálculo completo do grafo de navegação. A segunda diminui a quantidade de nós adicionados ao grafo. Finalmente, exemplos práticos são resolvidos para facilitar o entendimento dos algoritmos. O material aqui exposto foi largamente baseado nos trabalhos de (KAESS, 2011b, 2012; INDELMAN, 2013).

6.1 FRONT-END: PRÉ-INTEGRAÇÃO

Tipicamente as unidades de medidas iniciais (IMU) operam com taxas superiores a 100Hz. Isso quer dizer que nós são adicionados ao grafo de navegação a cada 0.01s. Em outras palavras, haverá 100 nós no grafo em apenas 1s. Mesmo com a abordagem incremental, sistemas robóticos com restrições de processamento e memória, podem não conseguir calcular as estimativas em tempo real. Para resolver o problema, diversas soluções foram propostas. O trabalho de (LATIF, 2013) reduz o grafo pela divisão da trajetória do robô usando segmentos de linha. Os autores se baseiam no fato de que os robôs tendem a mover-se em linha reta em muitos trechos de suas trajetórias. Dessa forma, nesses trechos, a trajetória pode ser substituída por um único segmento de reta. (GRISETTI, 2010) propõe a partição do grafo em diferentes níveis de hierarquia, onde os níveis mais baixos representam o grafo completo e os níveis mais altos representam amostragens dos níveis mais baixos. A otimização começa nos níveis mais baixos e são propagadas para os níveis mais altos em função de um limiar de precisão escolhido. Já o trabalho de (SUGER, 2014) adota uma metodologia de dividir-para-conquistar que também subdivide o grafo em vários subgrafos de tamanhos menores. Em seguida, os subgrafos são resolvidos recursivamente até atingir a precisão desejada. Diferentemente desses trabalhos e mais próximo do nosso problema, utilizamos a técnica de pré-integração. Essa técnica foi inicialmente desenvolvida por (LUPTON, 2012) onde as diversas medições de uma unidade de medida inercial são integradas por um pequeno período de tempo e só são utilizadas quando medições adicionais são recebidas. O trabalho de (INDELMAN, 2013)

adapta a abordagem anterior para o modelo gráfico dos grafos-fatores, sendo portanto, de fácil utilização no problema da suavização incremental (ou batch). Em resumo, ao invés de adicionar nós ao grafo na taxa de operação da IMU, pré-integramos as suas medições até o momento em que dados de sensores mais lentos são recebidos. O Algoritmo 1 apresenta o pseudocódigo para a pré-integração. A FIG. 6.1 ilustra um grafo-fator com e sem a pré-integração.

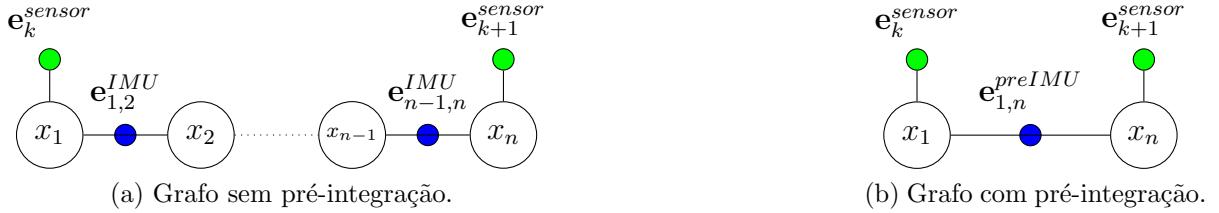


FIG. 6.1: Grafo-fator sem (a) e com pre-integração. A técnica diminui a quantidade de nós adicionados ao grafo.

Algoritmo 1: Pré-integração das medições da IMU $[f_j, w_j]$. (INDELMAN, 2013).

Entrada: Medições da IMU: $[f_j, w_j]$ (forças específicas, velocidades angulares)

Parâmetros de calibração (bias estático dos acelerômetros e girômetros)

- 1: Compense as medições da IMU com os parâmetros de calibração c_i
- 2: Posição: $\Delta p_{i \rightarrow j+1}^{b_i} = \Delta p_{i \rightarrow j}^{b_i} + \Delta v_{i \rightarrow j}^{b_i} \Delta t$
- 3: Velocidade: $\Delta v_{i \rightarrow j+1}^{b_i} = \Delta v_{i \rightarrow j}^{b_i} + R_{b_j}^{b_i} f_j \Delta t$
- 4: Orientação: $R_{b_{j+1}}^{b_i} = R_{b_j}^{b_i} \text{Expmap}(w_j)$

Saída: Pré-integrações: $\Delta p_{i \rightarrow j+1}^{b_i}$, $\Delta v_{i \rightarrow j+1}^{b_i}$, $R_{b_{j+1}}^{b_i}$

Nesse caso adotou-se uma simples integração de Euler. Integrações de ordem maior podem ser facilmente adaptadas. O intervalo de integração está compreendido entre os instantes i e j . As integrações são realizadas em relação ao corpo da aeronave no instante i , sendo representado pelo sobrescrito b_i . A rotação $R_{b_j}^{b_i}$ transforma o referencial do corpo no instante j para o referencial do corpo no instante i . O operador *Expmap* calcula a variação angular em função da velocidade angular. Detalhes podem ser encontrados em (PLUMBLEY, 2004; GRASSIA, 1998).

O problema da pré-integração é que a taxa de adição fica constrita ao intervalo de operação dos outros sensores. Assim, se um sensor adicional como um GPS trabalhar com taxa de 5Hz, as estimativas serão calculadas com essa taxa. Ou seja, a cada 0.2s.

Para robôs terrestres que se movimentam com baixa velocidade, isso não chega a ser um problema. No caso de um VANT que realiza movimentos muitos mais rápidos e agressivos, essa taxa é lenta para efeitos de controle. O controle é a parte do piloto automático responsável por acionar o motor e as superfícies de controle de maneira a ajustar a trajetória da aeronave. Se a taxa for lenta, o controle irá realizar ajustes mais bruscos, o que pode acarretar perda de estabilidade e até mesmo a queda da aeronave. Percebemos isso nas simulações discutidas no Capítulo 8. A solução é calcular as previsões na taxa da IMU e enviar para o piloto automático para a realização do controle. Apesar de não haver a etapa de correção, como o intervalo de pré-integração é pequeno, o erro acumulado também será pequeno. O Algoritmo 2 mostra como o cálculo é feito. Em resumo, dado que temos as estimativas anteriores utilizamos os deltas obtidos com o Algoritmo 1 para projetarmos no tempo atual a estimativa requerida, compensando os efeitos da gravidade e da aceleração da Terra.

Algoritmo 2: Previsão do fator IMU. (INDELMAN, 2013).

Entrada: Pré-integrações: $\Delta p_{i \rightarrow j+1}^{b_i}$, $\Delta v_{i \rightarrow j+1}^{b_i}$, $R_{b_{j+1}}^{b_i}$

Estimativas anteriores: p_i, v_i, R

$$1: \text{Posição: } p_j^{L_j} = R_{L_i}^{L_j} p_i^{L_i} + R_{b_i}^{L_i} \Delta p_{i \rightarrow j}^{b_i} + v_i^{L_i} (t_j - t_i) + [\frac{1}{2} g^{L_i} - 2[\omega_{iL_i}^{L_i}]_\times v_i^{L_i}] (t_j - t_i)^2$$

$$2: \text{Velocidade: } v_j^{L_j} = R_{L_i}^{L_j} v_i^{L_i} + R_{b_i}^{L_i} \Delta v_{i \rightarrow j}^{b_i} + [g^{L_i} - 2[\omega_{iL_i}^{L_i}]_\times v_i^{L_i}] (t_j - t_i)$$

$$3: \text{Orientação: } R_{b_j}^{L_j} = R_{L_i}^{L_j} R_{b_i}^{L_i} \text{Expmap}(\Delta\phi)$$

$$\text{onde } \Delta\phi = \text{Logmap}(R_{b_j}^{b_i}) - R_{L_i}^{b_i} \omega_{iL_i}^{L_i} (t_j - t_i)$$

Saída: Previsões: $p_j^{L_j}, v_j^{L_j}, R_{b_j}^{L_j}$

A matrix $R_{b_i}^{L_i}$ representa a rotação do referencial do corpo da aeronave para o referencial inercial local e a gravidade é representada por g^{L_i} .

6.2 BACK-END: SUAVIZAÇÃO INCREMENTAL

Como vimos anteriormente, a fusão dos dados dos sensores utilizou técnicas de otimização via fatoração de matrizes para encontrar a solução requerida. Nesta seção, decreveremos como solucionar a fusão operando diretamente com modelos gráficos probabilísticos sem a necessidade de converter o grafo-fator em uma matriz para, em seguida, aplicar os métodos da álgebra linear.

O primeiro passo do procedimento é transformar o grafo-fator em uma rede Bayesiana. Para isso aplica-se o Algoritmo 3. Nesse algoritmo as variáveis são eliminadas uma a uma. Como resultado temos uma rede Bayesiana equivalente a uma matriz triangular superior. Em outras palavras o algoritmo de eliminação de variáveis é utilizado para substituir a fatoração de matrizes. Para visualizar isso, repare que a matriz dos Jacobianos A associada ao grafo-fator da FIG. 6.2 (a) foi transformada na matriz triangular superior R associada à rede de Bayes da FIG. 6.2 (b).

Algoritmo 3: (Grafo-fator para rede Bayesiana) Transforma um grafo-fator em uma rede Bayesiana mediante a eliminação de todas as variáveis x_j dada uma ordem de eliminação. Adaptado de (KAESS, 2011a)

Entrada: Grafo-fator, variáveis x_j em ordem de eliminação

Saída: Rede Bayesiana

para cada x_j **faça**

1. Remova todos os fatores $f_i(x_i)$ que são adjacentes a x_j .
2. Defina os separadores S_j como sendo todas as variáveis envolvendo os fatores adjacentes, exceto x_j .
3. Calcule a densidade conjunta de x_j e S_j como sendo o produto desses fatores:

$$f_{conj}(x_j, S_j) = \prod_i f_i(x_i)$$

4. Use a regra da multiplicação $P(X, Y) = P(X|Y).P(Y)$ para obter a probabilidade condicional:

$$f_{conj}(x_j, S_j) = P(x_j|S_j).f_{novo}(S_j)$$

5. Adicione a probabilidade condicional $P(x_j|S_j)$ na rede Bayesiana e o novo fator $f_{novo}(S_j)$ de volta ao grafo-fator.

fim

Finda a eliminação de todas as variáveis, a densidade da rede Bayesiana é definida como o produto das probabilidades condicionais resultante de cada passo do algoritmo anterior:

$$P(X) = \prod_j P(x_j|S_j) \tag{6.1}$$

onde S_j é o separador de x_j , isto é, o conjunto das variáveis que estão diretamente

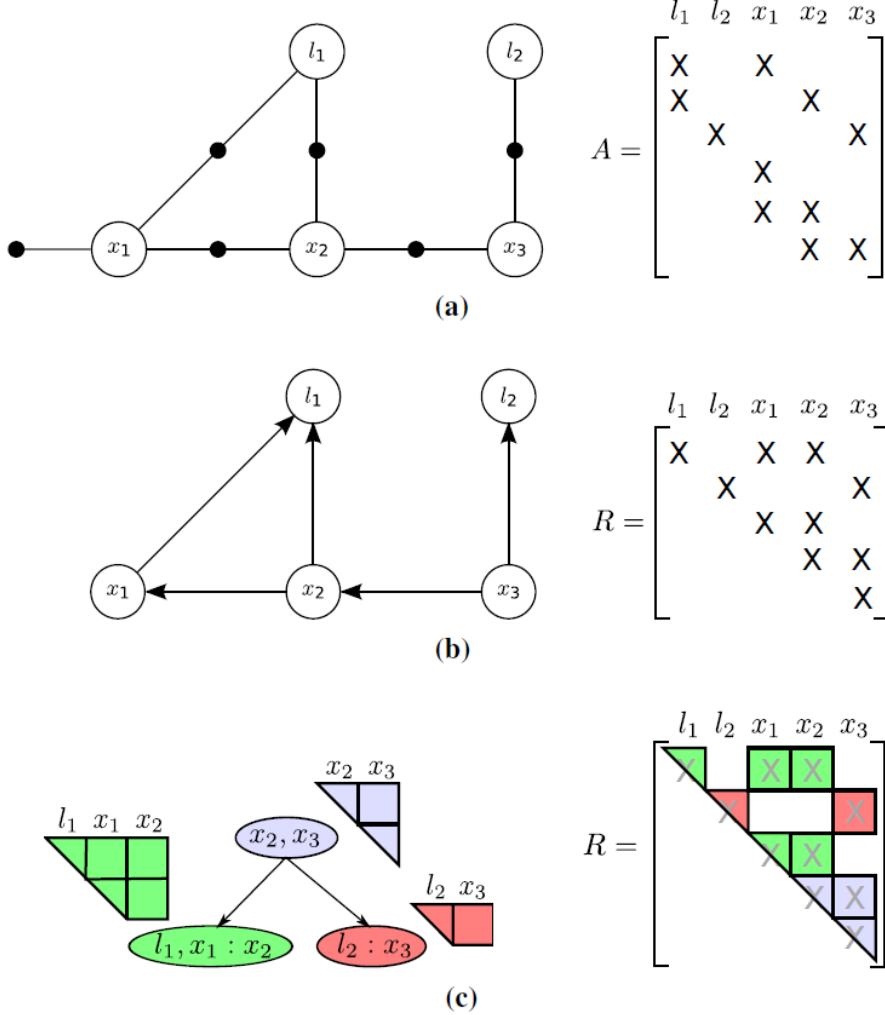


FIG. 6.2: (a) O grafo-fator e sua matriz Jacobiana A para um exemplo hipotético. (b) A rede Bayesiana triangulada e a matriz de informação R resultante da ordem de eliminação l_1, l_2, x_1, x_2, x_3 . (c) A árvore de Bayes e seus cliques (KAESS, 2011a).

conectadas à x_j por um fator. Em resumo, o produto dos fatores $\prod_i f_i(x_i)$ foi convertido em um produto de condicionais $\prod_j P(x_j|S_j)$ equivalente.

Outro fato interessante é que a rede Bayesiana resultante é triangulada, ou seja, podemos convertê-la em uma árvore, onde as operações incrementais serão fáceis. Essa estrutura foi chamada de árvore de Bayes por (KAESS, 2011a) sendo a maior contribuição de seu trabalho. Trata-se de uma árvore direcionada onde os nós representam os cliques C_k da rede Bayesiana. Isso quer dizer que esta nova estrutura codifica uma densidade de probabilidade fatorada

$$P(X) = \prod_k P(F_k|S_k) \quad (6.2)$$

onde, S_k é definido como o separador (a interseção $C_k \cap \Pi_k$ entre o clique C_k e seu clique pai Π_k), e a variável frontal F_k são as variáveis restantes ($F_k \triangleq C_k \setminus S_k$). A árvore Bayesiana utiliza a notação $C_k = F_k : S_k$ para representar seus nós. O Algoritmo 4 mostra como a conversão é feita e a FIG. 6.2 (c) mostra a árvore Bayesiana resultante.

Algoritmo 4: (Rede Bayesiana para árvore Bayesiana). Gera uma árvore Bayesiana a partir de uma rede Bayesiana triangulada. Adaptado de (KAESS, 2011a).

Entrada: Rede Bayesiana

Saída: Árvore Bayesiana

para cada probabilidade condicional $P(X_i|S_i)$ da rede Bayesiana, na ordem inversa da eliminação **faça**

```

se não há nó pai ( $S_i = \{\}$ ) então
| adicione um novo clique na raiz com as variáveis frontais  $X_i$ 
senão
| procure um clique  $C_p$  que contenha a primeira variável eliminada  $S_i$  como
| uma variável frontal;
| se  $F_p \cup S_p = S_i$ , isto é, as variáveis frontais e separadoras do clique  $C_p$  são
| iguais às variáveis separadoras  $S_i$  então
| | adicione a probabilidade condicional  $P(X_i|S_i)$  no clique  $C_p$ 
| senão
| | crie um novo clique  $C'$  como filho do clique  $C_p$ 
| fim
fim
fim

```

6.2.1 INFERÊNCIA INCREMENTAL

O aspecto fundamental da estrutura anterior é que a otimização ou a fusão dos dados dos sensores pode ser realizada de forma incremental com uma simples atualização da árvore Bayesiana. Isso quer dizer que a cada nova medição de um sensor somente parte da árvore será afetada, sendo o cálculo realizado somente nessa parte. Informalmente, a parte afetada da árvore Bayesiana é convertida novamente em um grafo-fator, o novo fator da medição é adicionado ao grafo-fator resultante e então aplicamos o Algoritmo 3 e o Algoritmo 4. A árvore Bayesiana formada é então mesclada com as partes inalteradas da árvore Bayesiana original obtendo assim a rede Bayesiana atualizada. O Algoritmo

5 apresenta o procedimento de atualização incremental e a FIG. 6.3 ilustra o processo quando um novo fator é adicionado entre as variáveis x_1 e x_3 . Somente a subárvore da esquerda é afetada e atualizada.

Algoritmo 5: (Atualização da árvore Bayesiana). Atualiza a árvore com os novos fatores F_{novo} . Adaptado de (KAESS, 2011a)

Entrada: Árvore Bayesiana \mathcal{T} , novos fatores \mathcal{F}_{novo}

Saída: Árvore Bayesiana atualizada \mathcal{T}_{novo}

início

1. Encontre todos os cliques C_{topo} que contenham as variáveis afetadas por \mathcal{F}_{novo} .
2. Remova os cliques C_{topo} incluindo seus parentes até o nó raiz e converta a subárvore resultante num grafo-fator \mathcal{F}_{topo} .
3. Armazene a subárvore restante em \mathcal{T}_{velha} .
4. Adicione os novos fatores \mathcal{F}_{novo} ao grafo-fator \mathcal{F}_{topo} ;
5. Encontre uma nova ordem de eliminação para \mathcal{F}_{novo} .
6. Elimine as variáveis de \mathcal{F}_{novo} , conforme o Algoritmo 3.
7. Crie a árvore Bayesiana \mathcal{T}' conforme o Algoritmo 4.
8. Retorne a árvore atualizada \mathcal{T}_{novo} juntando a árvore \mathcal{T}_{velha} com a árvore \mathcal{T}' ;

fim

Assim, incorporar novas medidas de diferentes sensores, torna-se uma questão de conectar os fatores definidos por essas medidas às variáveis apropriadas do grafo-fator. Da mesma forma, se um sensor torna-se indisponível, devido a perda de sinal ou falha do mesmo, o sistema simplesmente se abstém de adicionar os fatores associados. Não é necessário nenhum procedimento especial. Essa facilidade permite representar o problema da localização com várias configurações de sensores que operam com diferentes taxas. A FIG. 6.4 apresenta uma estrutura simplificada de um possível grafo de navegação para o conjunto de sensores embarcado no nosso VANT. Note que cada sensor (nós coloridos) possui uma taxa de operação diferente. A odometria ou o movimento é representado pelos nós pretos. As variáveis de estado estão contidas nos nós x_i .

Exemplo 6.1. Reconsidere o EX. 4.1. Suponha agora que estamos no tempo $t = 3s$ e em seguida passamos para o tempo $t = 4s$. A FIG. 6.5 ilustra o grafo-fator para o exemplo, onde as partes afetadas estão tracejadas.

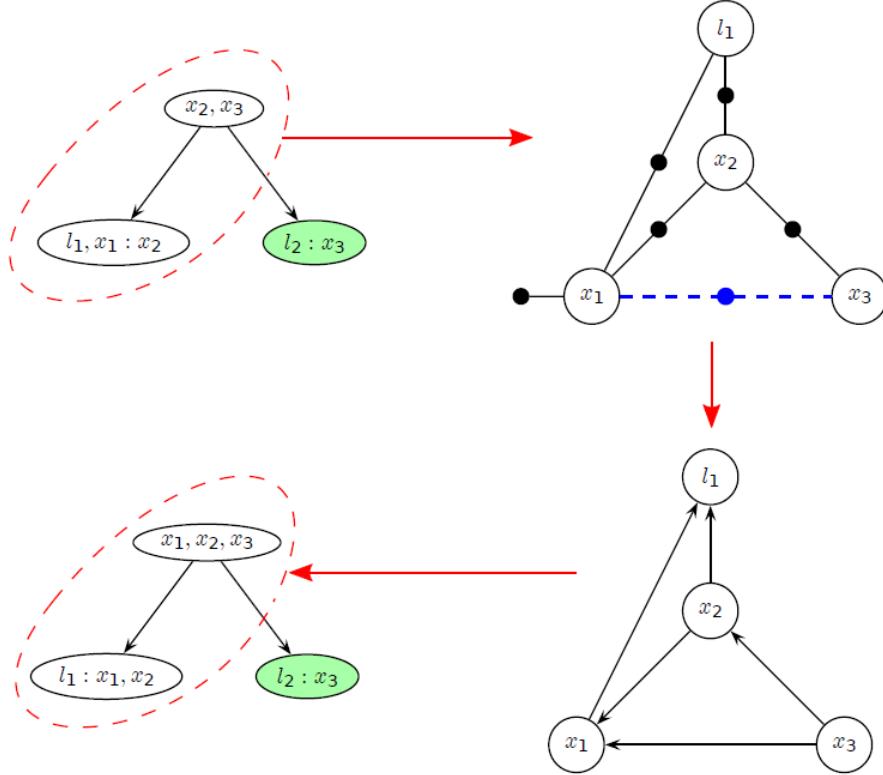


FIG. 6.3: Atualização da árvore Bayesiana da figura 6.2(c) quando um novo fator entre x_1 e x_3 (em azul) é adicionado. A subárvore da direita não é afetada. (KAESS, 2011a)

Erro das medições: $h(x_k) - z_k$

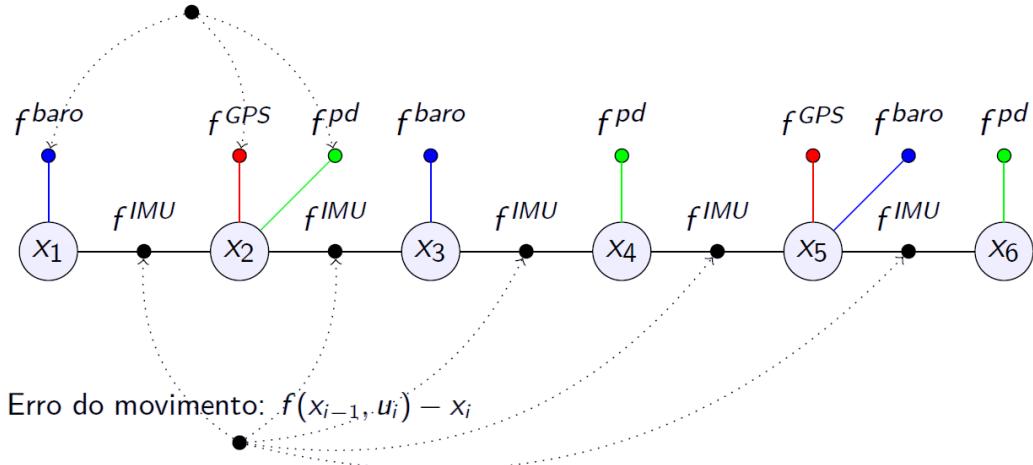


FIG. 6.4: Representação ilustrativa do grafo de navegação do VANT. Os sensores (nós coloridos) operam com taxas diferentes e vão sendo adicionados ao grafo, onde f^{pd} é o fator do sensor de pressão diferencial, f^{baro} é o fator do barômetro, f^{GPS} é o fator do GPS e f^{IMU} é o fator do movimento.

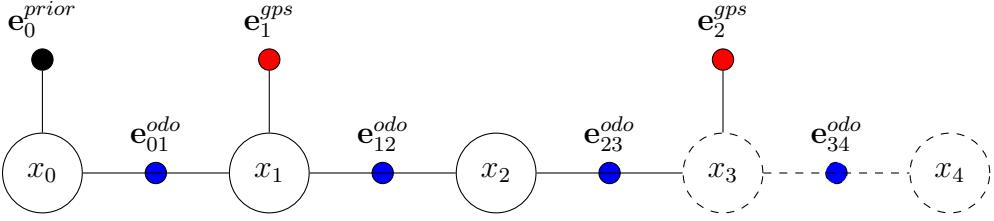


FIG. 6.5: Grafo-fator para o EX. 6.1. No instante $t = 4s$ recebemos uma nova medição. As partes afetadas estão tracejadas.

Não iremos ver a solução incremental completa, ou seja, aplicarmos os algoritmos nos tempos $t = 1, 2, 3$ e $4s$, o que seria trabalhoso e desnecessário. Por isso, vamos considerar que estamos no tempo $t = 3s$. Logo, devemos recriar a rede Bayesiana (Algoritmo 3) e a árvore de Bayes (Algoritmo 4) no tempo $t = 3s$. Em seguida executaremos a etapa incremental.

O primeiro passo será criar a rede Bayesiana. Como o Algoritmo 3 precisa de uma ordem de eliminação, usaremos a ordem (x_0, x_1, x_2, x_3) . Note que a ordem não é dada, mas calculada pelo algoritmo por uma heurística conhecida como COLAMD (COLumn Approximate Minimum Degree - algoritmo de eliminação que baseia-se na evolução do grau dos vértices, eliminando a cada passo o vértice de menor grau, fazendo com que seus vizinhos se tornem adjacentes) (DAVIS, 2004). O uso de uma heurística para o cálculo da ordem de eliminação é devido ao problema ser NP-completo (YANNAKAKIS, 1981). A ordem de eliminação é importante pois afeta a performance, uma vez que, quanto melhor a ordem, mais esparsa se torna a matriz Jacobiana. Para uma introdução sobre o assunto e comparativos entre diversas heurísticas de ordenamento de variáveis citamos (AGARWAL, 2012).

Antes de executar o algoritmo vamos precisar dos Jacobianos e das recíprocas das raízes quadradas das matrizes de covariância. Vimos essa forma de representação na Seção 4.1. O cálculo dos Jacobianos já foi realizado no EX. 4.1. Repetimos aqui os Jacobianos em conjunto com a recíproca da raiz quadrada das covariâncias:

$$\mathbf{J} = \begin{array}{c|ccccc} & x_0 & x_1 & x_2 & x_3 & x_4 \\ \hline \mathbf{e}_{01}^{odo} & -1 & 1 & 0 & 0 & 0 \\ \mathbf{e}_{12}^{odo} & 0 & -1 & 1 & 0 & 0 \\ \mathbf{e}_{23}^{odo} & 0 & 0 & -1 & 1 & 0 \\ \mathbf{e}_{34}^{odo} & 0 & 0 & 0 & -1 & 1 \\ \mathbf{e}_0^{prior} & 1 & 0 & 0 & 0 & 0 \\ \mathbf{e}_1^{gps} & 0 & 1 & 0 & 0 & 0 \\ \mathbf{e}_2^{gps} & 0 & 0 & 0 & 1 & 0 \end{array}$$

$$\Sigma_{odo}^{-1/2} = \frac{1}{\sqrt{1^2}} = 1$$

$$\Sigma_{prior}^{-1/2} = \frac{1}{\sqrt{0.1^2}} = 10$$

$$\Sigma_{gps}^{-1/2} = \frac{1}{\sqrt{0.5^2}} = 2$$

Seguindo passo a passo o Algoritmo 3:

Etapa 1: Eliminar a variável x_0 .

- a) Remover os fatores x_j adjacentes a x_0 : $x_j = \{e_0^{prior}(x_0), e_{01}^{odo}(x_0, x_1)\}$
- b) Definir os separadores S_j : $S_j = \{x_1\}$
- c) Calcular a densidade conjunta: $f_{conj}(x_j, S_j) = f_{conj}(x_0, x_1) = \prod_i f_i(x_i)$. Nós já vimos que podemos transformar essa probabilidade conjunta em $\|A_j \Delta - a_j\|_2^2$, logo:

$$\mathbf{A}_0 = \begin{array}{c|cc} & x_0 & x_1 \\ \hline \Sigma_0^{-1/2} J \mathbf{e}_0^{prior} & 10 & 0 \\ \Sigma_{01}^{-1/2} J \mathbf{e}_{01}^{odo} & -1 & 1 \end{array} \quad \mathbf{a}_0 = \begin{array}{c|c} -\Sigma_0^{-1/2} \mathbf{e}_0^{prior} & x_0 - z_0^{prior} \\ -\Sigma_{01}^{-1/2} \mathbf{e}_{01}^{odo} & -x_0 + x_1 - u_{01}^{odo} \end{array} = \begin{bmatrix} 0.0436 \\ 10.0414 \end{bmatrix}$$

- d) Calcule a probabilidade condicional $f_{conj}(x_j, S_j) = P(x_j | S_j) \cdot f_{novo}(S_j)$

$$[\mathbf{A}_0 \quad \mathbf{a}_0] = \mathbf{Q} \begin{bmatrix} \mathbf{R} & \mathbf{A}_S & \mathbf{d} \\ 0 & \mathbf{A}_{novo} & \mathbf{a}_{novo} \end{bmatrix} = \mathbf{Q} \begin{bmatrix} x_0 & x_1 \\ -10.0499 & 0.0995 & 0.5653 \\ 0 & 0.9950 & 10.0350 \end{bmatrix}$$

A probabilidade condicional é dada por:

$$\begin{aligned} P(x_j | S_j) = P(x_0 | x_1) &= \exp \left(-\frac{1}{2} \|\mathbf{R}x_0 - (\mathbf{d} - \mathbf{A}_S x_1)\|_2^2 \right) \\ &= \exp \left(-\frac{1}{2} \| -10.0499x_0 - (0.5653 - 0.0995x_1) \|_2^2 \right) \end{aligned}$$

E o novo fator é dado por:

$$f_1^{novo}(x_1) = \exp\left(-\frac{1}{2} \|\mathbf{A}_{\text{novo}}x_1 - \mathbf{a}_{\text{novo}}\|_2^2\right) = \exp\left(-\frac{1}{2} \|0.995x_1 - 10.0350\|_2^2\right)$$

A FIG. 6.6 ilustra o grafo após a eliminação da variável x_0 . Nesse grafo, o novo fator resultante do processo de eliminação está representado no cor amarela. Os vértices vermelhos representam os fatores com as informações das medições do GPS. Os azuis, contém as informações da odometria (modelo do movimento função dos dados do acelerômetro e do girômetro).

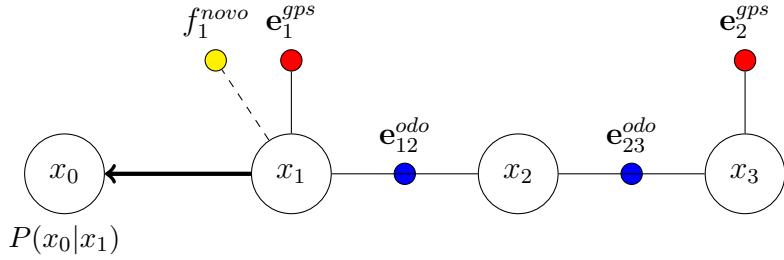


FIG. 6.6: Grafo-fator apóis a eliminação da variável x_0 . Trata-se de um modelo híbrido entre um grafo-fator com uma rede Bayesiana. O novo fator resultante da eliminação está representado em amarelo.

Etapa 2: Eliminar a variável x_1 .

- a) $x_j = \{f_1^{novo}(x_1), e_1^{gps}(x_1), e_{12}^{odo}(x_1, x_2)\}$
- b) $S_j = \{x_2\}$
- c) Calcular a densidade conjunta: $f_{conj}(x_j, S_j) = f_{conj}(x_1, x_2)$

$$\mathbf{A}_1 = \begin{matrix} & \begin{matrix} x_1 & x_2 \end{matrix} \\ \begin{matrix} f_1^{novo} \\ \Sigma_1^{-1/2} J e_1^{gps} \\ \Sigma_{12}^{-1/2} J e_{12}^{odo} \end{matrix} & \begin{bmatrix} 0.9950 & 0 \\ 2 & 0 \\ -1 & 1 \end{bmatrix} \end{matrix} \quad \mathbf{a}_1 = \begin{matrix} & \begin{matrix} x_1 & x_2 \end{matrix} \\ \begin{matrix} \mathbf{a}_{\text{novo}} \\ x_1 - z_1^{gps} \\ -\Sigma_{12}^{-1/2} e_{12}^{odo} \end{matrix} & \begin{bmatrix} \mathbf{a}_{\text{novo}} \\ x_1 - z_1^{gps} \\ -x_1 + x_2 - u_{12}^{odo} \end{bmatrix} \end{matrix} = \begin{bmatrix} 10.0350 \\ 19.6272 \\ 9.2658 \end{bmatrix}$$

- d) Calcule a probabilidade condicional $f_{conj}(x_j, S_j) = P(x_j|S_j) \cdot f_{novo}(S_j)$

$$[\mathbf{A}_1 \quad \mathbf{a}_1] = \mathbf{Q} \begin{bmatrix} & \begin{matrix} x_1 & x_2 \end{matrix} \\ & \begin{bmatrix} -2.4475 & 0.4086 & -16.3327 \\ 0 & -0.9127 & -17.4634 \\ 0 & 0 & 0.2422 \end{bmatrix} \end{bmatrix}$$

A probabilidade condicional é dada por:

$$P(x_j|S_j) = P(x_0|x_1) = \exp\left(-\frac{1}{2} \| -2.4475x_1 - (-16.3327 - 0.4086x_2) \|_2^2 \right)$$

E o novo fator é dado por:

$$f_2^{novo}(x_2) = \exp\left(-\frac{1}{2} \| -0.9127x_2 + 17.4634 \|_2^2 \right)$$

A FIG. 6.7 ilustra o grafo após a eliminação da variável x_1 . O novo fator f_2^{novo} está representado em amarelo. Novamente, o fator vermelho representa o GPS e o azul a odometria calculado com os dados da unidade de medida inercial.

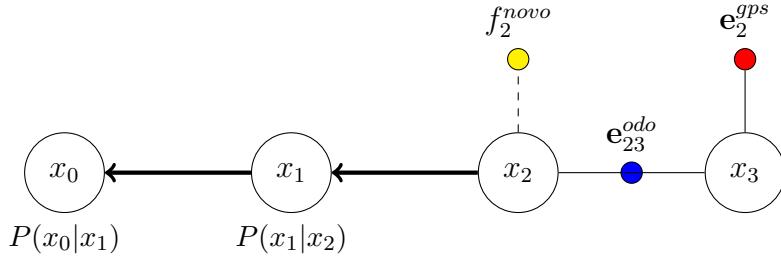


FIG. 6.7: Grafo híbrido apóis a eliminação da variável x_1 . O novo fator resultante da eliminação está representado em amarelo.

Etapa 3: Eliminar a variável x_2 .

- a) $x_j = \{f_2^{novo}(x_2), e_{23}^{odo}(x_2, x_3)\}$
- b) $S_j = \{x_3\}$
- c) Calcular a densidade conjunta: $f_{conj}(x_j, S_j) = f_{conj}(x_2, x_3)$

$$\mathbf{A}_2 = \begin{matrix} & x_2 & x_3 \\ f_2^{novo} & \left[\begin{array}{cc} -0.9127 & 0 \\ -1 & 1 \end{array} \right] & \end{matrix} \quad \mathbf{a}_2 = \begin{matrix} & \mathbf{a}_{novo} \\ -\Sigma_{23}^{-1/2} \mathbf{e}_{23}^{odo} & \left[\begin{array}{c} -x_2 + x_3 - u_{23}^{odo} \end{array} \right] \end{matrix} = \begin{matrix} & -17.4634 \\ & 9.9692 \end{matrix}$$

- d) Calcule a probabilidade condicional $f_{conj}(x_j, S_j) = P(x_j|S_j) \cdot f_{novo}(S_j)$

$$[\mathbf{A}_2 \quad \mathbf{a}_2] = \mathbf{Q} \left[\begin{array}{ccc} 1.3539 & -0.7386 & 4.4093 \\ 0 & 0.6741 & 19.6192 \end{array} \right]$$

A probabilidade condicional é dada por:

$$P(x_j|S_j) = P(x_2|x_3) = \exp\left(-\frac{1}{2} \|1.3539x_2 - (-0.7386 - 4.4093x_3)\|_2^2\right)$$

E o novo fator é dado por:

$$f_3^{novo}(x_3) = \exp\left(-\frac{1}{2} \|0.6741x_3 - 19.6192\|_2^2\right)$$

A FIG. 6.8 ilustra o grafo após a eliminação da variável x_2 . O novo fator f_3^{novo} está representado em amarelo. Ainda temos um dado de GPS em vermelho.

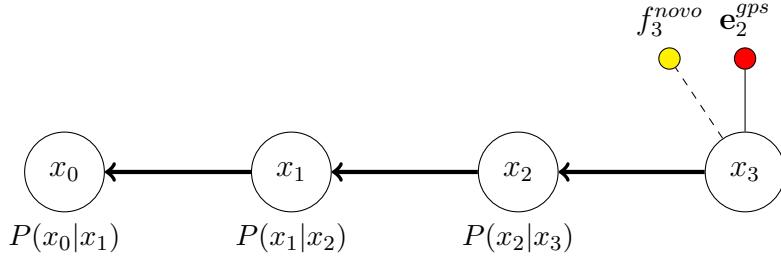


FIG. 6.8: Grafo híbrido apóis a eliminação da variável x_2 . O novo fator resultante da eliminação está representado em amarelo.

Etapa 4: Eliminar a variável x_3 .

- a) $x_j = \{f_3^{novo}(x_3), e_2^{gps}(x_3)\}$
- b) $S_j = \{\}$
- c) Calcular a densidade conjunta: $f_{conj}(x_j, S_j) = f_{conj}(x_2, x_3)$

$$\mathbf{A}_3 = \frac{f_2^{novo}}{\Sigma_2^{-1/2} J e_2^{gps}} \begin{bmatrix} 0.6741 \\ 2 \end{bmatrix} \quad \mathbf{a}_3 = \frac{\mathbf{a}_{novo}}{-\Sigma_2^{-1/2} \mathbf{e}_2^{gps}} \begin{bmatrix} \mathbf{a}_{novo} \\ x_3 - z_3^{gps} \end{bmatrix} = \begin{bmatrix} 19.6192 \\ 59.7636 \end{bmatrix}$$

- d) Calcule a probabilidade condicional $f_{conj}(x_j, S_j) = P(x_j|S_j) \cdot f_{novo}(S_j)$

$$[\mathbf{A}_3 \quad \mathbf{a}_3] = \mathbf{Q} \begin{bmatrix} -2.1105 & -62.8996 \\ 0 & 0.4967 \end{bmatrix}$$

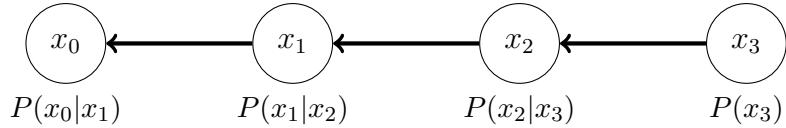


FIG. 6.9: Rede Bayesiana resultante da eliminação de todas as variáveis do grafo-fator do EX. 6.1. A probabilidade conjunta é $P(X) = P(x_0|x_1)P(x_1|x_2)P(x_2|x_3)P(x_3)$

A probabilidade condicional é dada por:

$$P(x_j|S_j) = P(x_2|x_3) = \exp\left(-\frac{1}{2} \| -2.1105x_3 + 62.8996 \|_2^2 \right)$$

A FIG. 6.9 ilustra o grafo apóis a eliminação da variável x_3 .

Depois de criada a rede Bayesiana, aplica-se o Algoritmo 4 para obter-se a seguinte árvore Bayesiana:

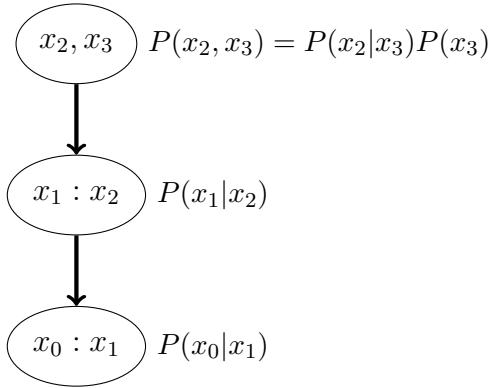


FIG. 6.10: Árvore Bayesiana resultante da aplicação do Algoritmo 4 na rede de Bayes da FIG. 6.9.

Repare que a probabilidade representada pela árvore Bayesiana é a mesma da rede de Bayes. Basta ver que $P(x_2, x_3) = P(x_2|x_3)P(x_3)$. Finalmente, no tempo $t = 4s$ recebemos uma nova informação, conforme já ilustrado na FIG. 6.5 do EX. 6.1. Nessa fase, aplicamos o Algoritmo 5. A etapa que merece atenção é aquela que encontra as partes afetadas da árvore. No nosso exemplo um novo fator $e_{34}^{odo}(x_3, x_4)$ é adicionado entre a variável já existente x_3 e uma nova variável x_4 . Nesse caso o único clique que contém uma dessas variáveis é o clique raiz C_{raiz} . Como este clique possui as variáveis x_2 e x_3 , o grafo-fator a ser calculado possuirá as variáveis x_2, x_3 e x_4 . Utilizando novamente os Algoritmos 1 e 2, as novas estruturas serão:

A vantagem desse método é que somente partes do grafo são recalculadas ao invés do grafo completo.

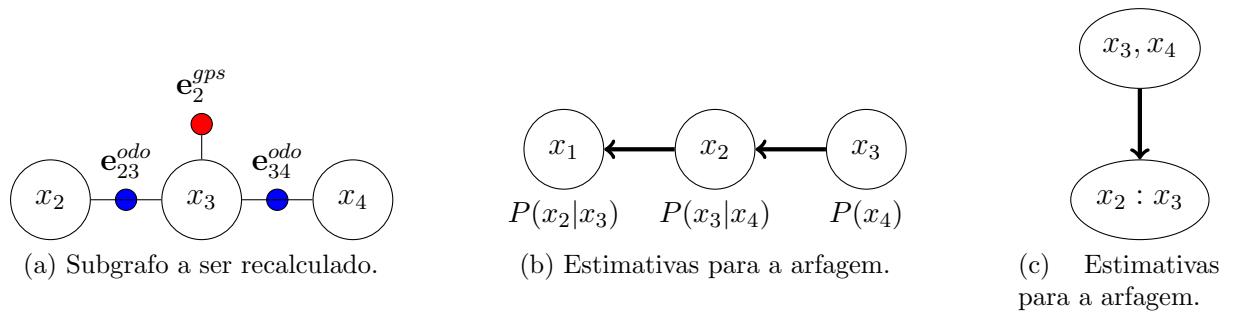


FIG. 6.11: Grafo afetado pela nova medição do tempo $t = 4$ e as correspondentes rede e árvore Bayesianas.

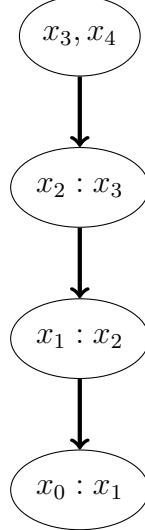


FIG. 6.12: Nova árvore Bayesiana após substituir o clique C_{raiz} anterior pela árvore da FIG. 6.11c

7 METODOLOGIA

Neste capítulo descrevemos a arquitetura elaborada para a simulação do voo, desde a escolha dos softwares e até as tecnologias necessárias para os experimentos reais.

7.1 HARDWARE-IN-THE-LOOP (HIL)

Antes de realizarmos qualquer experimento real, devemos validar de alguma forma a metodologia criada. Uma das maneiras mais usuais para esse intento são as simulações. E no caso específico dos VANTs, a técnica mais empregada para as simulações de voo, é conhecida como hardware-in-the-loop.

Nessa técnica, as informações contendo os dados dos diversos sensores ou dos estados da aeronave devem ser enviadas do simulador de voo para o hardware contendo o piloto automático, que em seguida envia o resultado dos controles da altitude e da posição de volta para o simulador num processo ininterrupto. Dessa forma, é possível simular a dinâmica de voo da aeronave sem a necessidade de realizar um voo real. Isso permite, então, o diagnóstico da performance, a viabilidade da solução e o rastreio de defeitos no conjunto de software e de hardware que compõem a solução completa de navegação.

Dois níveis de simulação de HIL serão abordados. No primeiro nível, os dados dos sensores são enviados para o piloto automático e no segundo, a estimativa dos estados previamente calculada pela solução de fusão de sensores (GOPPERT, 2011). Essa diferenciação é importante, uma vez que, usualmente, quando encontramos o termo hardware-in-the-loop, é quase sempre implícito o primeiro nível. Em nosso trabalho, faremos uso do segundo nível. Isso se deve ao fato da placa que contém os sensores e o microprocessador não possuir poder computacional suficiente para realizar a fusão de sensores proposta por este trabalho. Isso não representa uma desvantagem do método, pois as placas de navegação normalmente são dimensionadas para interfacear somente com o conjunto interno de sensores, e as demais funcionalidades são obtidas por placas de expansão, de onde é possível a comunicação e a interação com os outros componentes eletrônicos, motores e sensores externos de um sistema robótico.

7.1.1 HIL VIA DADOS DE SENSORES

Definimos HIL via dados de sensores como o envio de pacotes contendo os dados dos diversos sensores para o piloto automático com o intuito de emular a entrada do sensor no sistema de fusão sensorial e, em seguida, receber os pacotes com o resultado do controle da altitude e da posição. Esse é o nível de HIL mais conhecido, sendo bastante eficaz para a avaliação final do desempenho do sistema antes de um experimento real. A desvantagem desse método é que muitas vezes torna-se difícil isolar os erros do sistema de fusão ou do sistema de controle. Além disso, os requisitos de comunicação para emular todas as entradas dos sensores podem ser muito exigentes. Sensores como os giroscópios e os acelerômetros geralmente operam em taxas superiores à 100 Hz, o que pode sobrecarregar o envio dos dados via comunicações seriais, como normalmente é feito. A FIG. 7.2 ilustra o conceito.

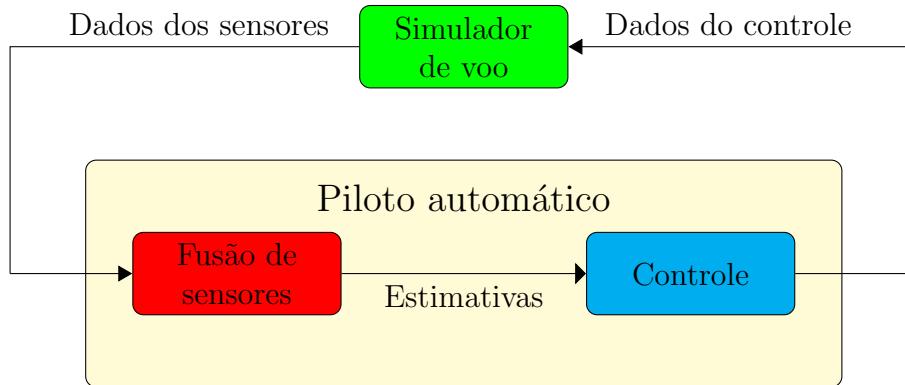


FIG. 7.1: HIL via dados de sensores. Os dados dos sensores são enviados para o piloto automático que realiza a fusão dos sensores.

7.1.2 HIL VIA DADOS DE ESTADOS

Definimos HIL via dados de estados como o envio de pacotes contendo os dados dos estados da aeronave diretamente para o piloto automático, emulando, assim, a saída do sistema de fusão sensorial e, em seguida, receber os pacotes do piloto automático com o resultado do controle. Esse nível de simulação, geralmente é utilizado para permitir o diagnóstico de problemas no sistema de controle, uma vez que podemos enviar os dados dos estados (do simulador) sem os erros provenientes do sistema de fusão sensorial, não havendo assim qualquer influência desse sistema. Neste caso assumimos que o sistema de fusão entrega o estado verdadeiro. Para contornar as desvantagens desse nível de HIL,

problemas como o *delay* e o ruído podem ser introduzidos pela simulação. A FIG. 7.2 ilustra esse nível.

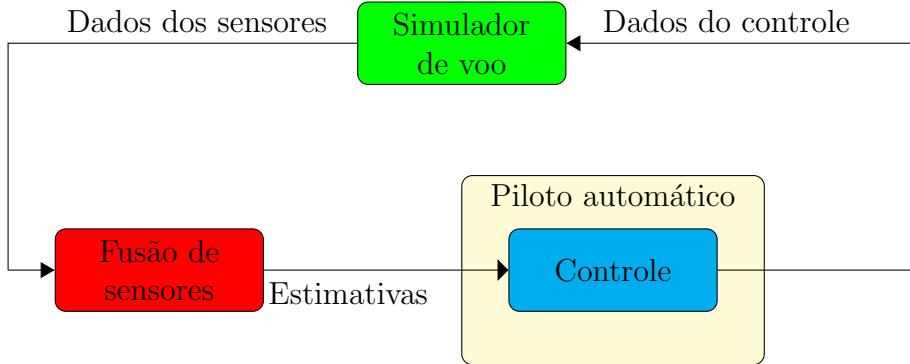


FIG. 7.2: HIL via dados de estados. Os dados dos sensores são enviados para um componente externo que irá realizar a fusão de sensores. O piloto automático recebe as estimativas prontas.

7.2 ARQUITETURA DA SIMULAÇÃO PROPOSTA

Em nossa arquitetura fazemos uso do HIL via dados de sensores, uma vez que o hardware que roda o piloto automático não possui poder computacional suficiente para a realização da fusão sensorial. Com isso, todo o processo de fusionamento é executado por um componente externo. No caso da simulação, optamos pelo uso de um plugin que se integra perfeitamente com o simulador de voo e realiza a troca de informações entre as outras partes do sistema mediante o uso do protocolo de comunicações MAVLink. A FIG. 7.3 ilustra a arquitetura proposta. Os detalhes de cada componente são apresentados nas próximas seções.

7.2.1 SIMULADOR DE VOO X-PLANE®

A integração de ambientes simulados com plataformas reais para testes e implementações de experimentos relacionados à movimentação de veículos autônomos apresenta grande relevância com a nossa abordagem. Dessa forma, optamos por simuladores consolidados como produtos utilizados tanto no mercado quanto no meio acadêmico. Nesse contexto, podemos representar cenários mais realistas para a validação do sistema de fusão sensorial e do voo autônomo. Dentre os simuladores de voo oferecidos no mercado, foi escolhido o X-Plane® por apresentar características mais relevantes ao desenvolvimento da aplicação proposta nesta dissertação.

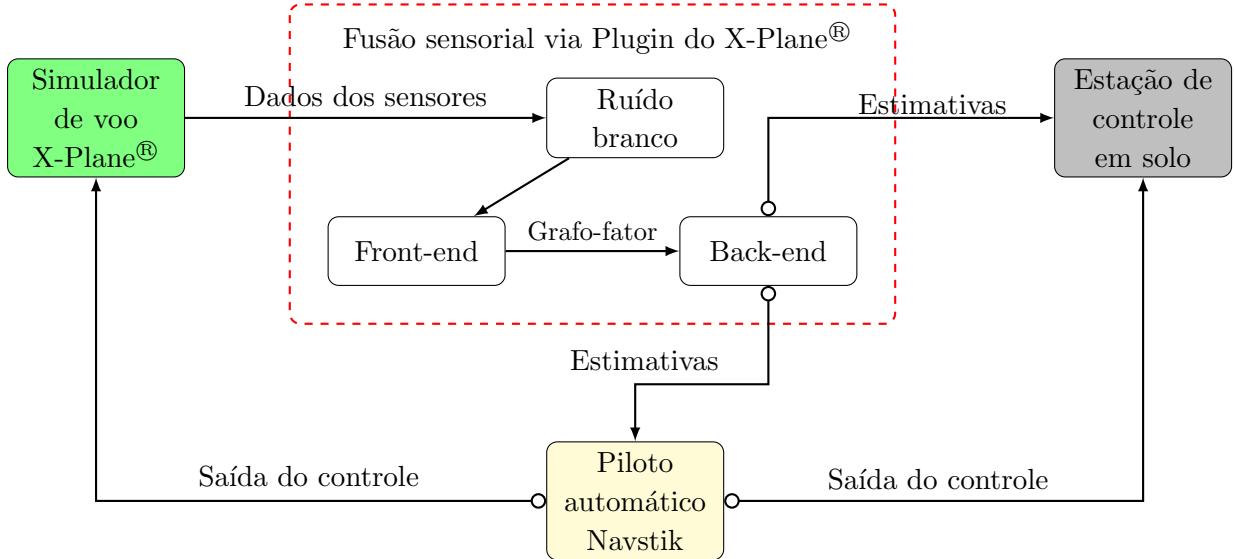


FIG. 7.3: Arquitetura de simulação. O plugin recebe os dados dos sensores provenientes do simulador de voo, adiciona ruído branco, gera o grafo-fator e finalmente, executa a suavização incremental. O protocolo MAVLink envia as estimativas para a ECS e o piloto automático que retorna as saídas do controle da atitude e posição.

O X-Plane[®], é um simulador de voo produzido pela Laminar Research (MEYER, 2014), que apresenta uma série de características que o destaca dos demais. Possui recursos de simulação física de voo, contendo um banco de dados de modelos de aeronaves e de cenários bastante amplo. Possui funcionalidades de geração aleatória de condições climáticas e cenários muito próximos aos de uma condição real de voo. Fatores estes, que o elencam como um dos poucos simuladores de voo a possuir homologação da Administração Federal de Administração, (do inglês: *Federal Aviation Administration - FAA*), para treinamento de pilotos. A FAA é a entidade governamental responsável pela regulamentação e supervisão de todos os assuntos relacionados à aviação civil nos Estados Unidos. No Brasil, esse papel cabe à Agência Nacional de Aviação Civil (ANAC).

Uma das características que torna o X-Plane[®] bastante útil em nossa metodologia é a possibilidade de extensão de suas funcionalidades por meio do desenvolvimento de plugins, mediante o uso de uma API (do inglês: *Application Program Interface*) em C++. Além disso, também é possível a obtenção e comunicação dos dados de voo através de datagramas UDP ou portas seriais. Optamos pela extensão via desenvolvimento de plugin, o que nos permitiu uma série de facilidades. A Seção 7.2.5 irá abordar os detalhes da implementação.

Finalmente, outra característica fundamental, é a possibilidade de edição e criação de modelos próprios de aeronaves expandindo ainda mais suas aplicações.

7.2.2 ESTAÇÃO DE CONTROLE EM SOLO QGROUNDCONTROL

A estação de controle em solo (ECS) é um componente crucial nas operações de voo de sistemas aéreos. Ela é responsável por disponibilizar todas as informações sobre a situação da aeronave e do voo para o controlador em solo. Para isso, ela necessita de um link de comunicações com a aeronave por onde essas informações trafegam. Esse link é conhecido como telemetria. Além disso, é possível o envio de comandos específicos para o controle da aeronave e para a execução de missões, normalmente definidas por meio de *waypoints* que irão determinar por onde a aeronave deverá passar, consequentemente realizando uma trajetória. O trabalho de (JOVANOVIC, 2008) descreve uma arquitetura geral de uma ECS para VANT com possibilidades de missões de reconhecimento e vigilância. Em (ANGONESE, 2013b) é proposta uma ECS capaz de controlar diversas aeronaves com funcionalidades de voo múltiplos em formação e desvio de obstáculos.

Para o nosso trabalho optamos pelo software QGroundControl (ZURICH, 2013), uma ECS em solo baseada no projeto de pesquisa PIXHAWK do grupo de computação visual e geométrica do departamento de computação científica do Instituto ETH Zurich. Dentre as características fundamentais para a sua escolha foram a integração com o protocolo de comunicação MAVlink e com o simulador de voo X-Plane®. Além disso, a estação conta com as seguintes funcionalidades:

- Suporte ao planejamento de missão por waypoints usando mapas 2D e 3D;
- Manipulação dos waypoints durante o voo;
- Telemetria em tempo real; e
- Gravação de logs.

A FIG. 7.4 demonstra a visão do operador durante a realização de uma missão. O QGroundControl conta com uma série de possibilidades visuais, facilitando o acompanhamento e o diagnóstico do voo e das missões.

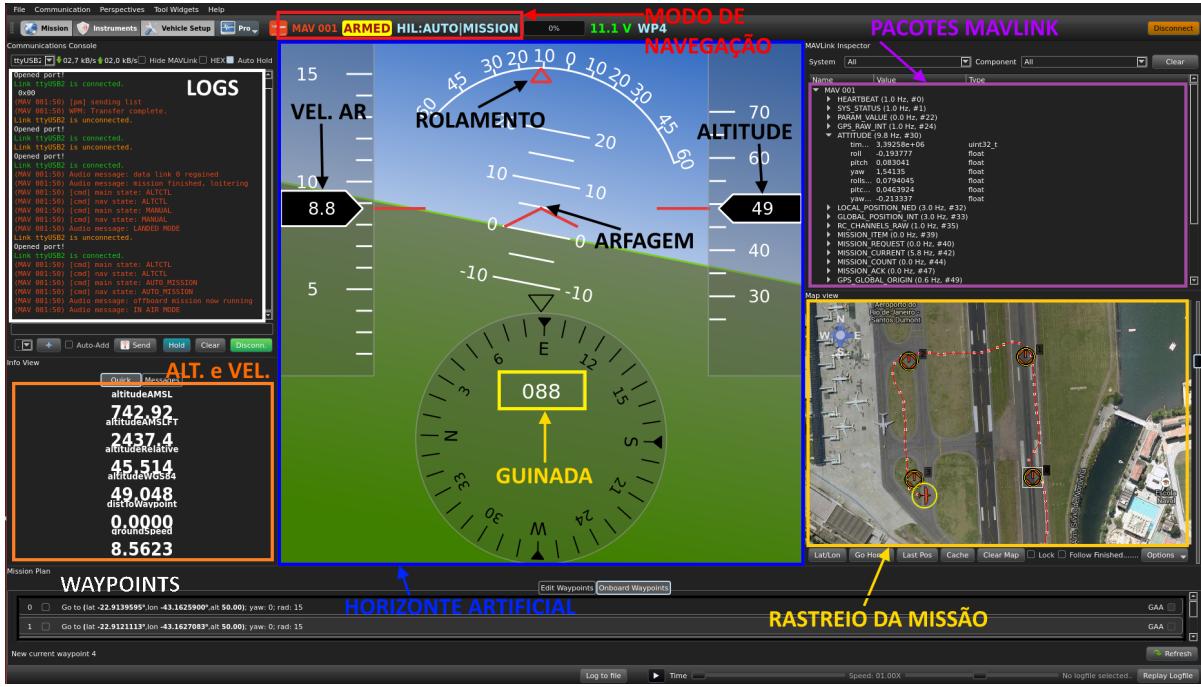


FIG. 7.4: Visão do operador durante a realização de uma missão pelo QGroundControl.

7.2.3 PROTOCOLO MAVLINK

O protocolo MAVLink (MEIER, 2009) desempenha um papel fundamental na execução da simulação (e também na operação da aeronave). Ele funciona como uma amalgama, ligando todos os componentes da arquitetura fornecendo uma especificação para direcionamento de mensagens. Sendo bastante leve e otimizado, o protocolo torna-se ideal para sistemas robóticos com restrições de desempenho. Ele possui apenas 8 bytes de conteúdo por pacote e permite o roteamento inter e intrassistemas. Além disso, devido à baixa sobrecarga computacional, as mensagens podem utilizar diversas camadas de transporte, como a UDP (*User Datagram Protocol*) e a UART (*Universal Asynchronous Receiver / Transmitter*). Atualmente o protocolo é utilizado por diversos pilotos automáticos: PIX-HAWK (MEIER, 2011), ArduPilotMega (APM, 2014), SLUGS (LIZARRAGA, 2013), SmartAP (SHILOV, 2014) e o PandaPilot (PANDAPILOT, 2014). Para um melhor entendimento de seu funcionamento é útil descrever a anatomia de seu pacote, e principalmente a estrutura da mensagem utilizada para trafegar as informações da estimativa dos estados.

7.2.4 ANATOMIA DO PACOTE

Os principais componentes de uma mensagem MAVLink são o cabeçalho (*header*), a identificação do sistema (*system ID*), a identificação da mensagem (*message ID*), e carga útil (*payload*). O cabeçalho é usado para classificar a mensagem como um pacote MAVLink, caso contrário será rejeitada. O *system ID* identifica o sistema que envia as mensagens (VANT A, VANT B, helicóptero, quadcóptero, robô terrestre) , enquanto que a *message ID* identifica o tipo de mensagem a ser enviada. Por exemplo, a mensagem mais comum a enviar é o *heartbeat* (*message ID* = 0) que é constantemente enviada para assegurar que a aeronave e a estação de controle em solo estão conectadas corretamente e podem se comunicar. Outros exemplos são o envio da altitude, da posição, de dados do gps, de dados da bateria. A estrutura dessas mensagens são definidas em formato XML e depois traduzidas automaticamente para uma linguagem de programação específica. Isso proporciona a portabilidade e a qualidade das mensagens no formato nativo definido. A documentação do protocolo apresenta uma lista extensa de mensagens pré-definidas. Caso nenhuma delas atenda aos requisitos da aplicação, é possível a especificação de mensagens customizadas. Finalmente, a carga útil é o conteúdo principal dentro dela. Esse campo pode conter dados como: o tipo de veículo, o modo de voo, os dados de posicionamento, os comandos para executar, os dados dos sensores, ou os dados dos estados. As mensagens são enviadas como pacotes de dados entre a estação de controle em solo e o VANT possibilitando o controle e acompanhamento em solo. A FIG. 7.5 apresenta a estrutura esquemática do pacote MAVLink e a TAB. 7.1 a descrição resumida de cada campo.

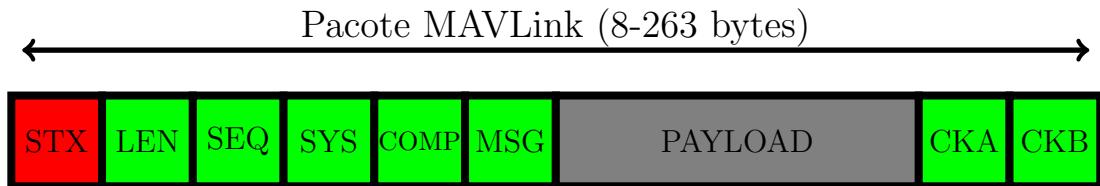


FIG. 7.5: Representação esquemática de um pacote MAVLink. (MEIER, 2011)

A partir dessas características verificamos que o tamanho mínimo do pacote é de 8 bytes (usado para os pacotes de confirmação sem payload) e o tamanho máximo é de 263 bytes, quando todo o campo de payload é utilizado.

Índice	Conteúdo	Valor	Descrição
STX	Início do pacote	0xFE	Indica o início de um novo pacote.
LEN	Tamanho do payload	0 - 255	Indica o tamanho do payload contido no pacote.
SEQ	Sequencial do pacote	0 - 255	Cada componente conta a sua sequência de envio. Permite detectar a perda de pacotes
SYS	System ID	1 - 255	Identificação do sistema que envia, permitindo diferenciar os sistemas robóticos da mesma rede.
COMP	Component ID	0 - 255	Identificação do componente que envia. Permite diferenciar diferentes componentes de um mesmo sistema robótico (IMU, piloto automático etc).
MSG	Message ID	0 - 255	Identificação da mensagem e do conteúdo do payload.
Payload	Dados	0 - 255	Dados da mensagem conforme seu tipo.
Checksum	Byte inicial e final		Conforme padrão ITU X.25/SAE AS-4.

TAB. 7.1: Descrição do pacote MAVLink

7.2.4.1 MENSAGEM HIL_STATE_QUATERNION

Além das mensagens usuais para o envio de comandos e de dados de telemetria, precisamos de uma mensagem específica para a realização da fusão dos sensores. A especificação do protocolo contém dois tipos de mensagens para essas situações: uma para o envio dos dados dos sensores (HIGHRES_IMU) e outra para o envio dos dados dos estados (HIL_STATE_QUATERNION). A mensagem HIGHRES_IMU carrega as medições dos acelerômetros, giroscópios, magnetômetros, barômetro e pressão diferencial. Em combinação com a mensagem GPS_RAW_INT, que contém as medições do GPS, temos todo o nosso conjunto de sensores sendo representados por essas mensagens. Já a mensagem HIL_STATE_QUATERNION carrega as informações dos estados da aeronave previamente fusionadas por algum processo externo (no nosso caso específico, o plugin desenvolvido para o simulador de voo ou uma placa *Computer-On-Module*). Nos casos mais gerais, havendo um bom link de comunicação, o computador que opera a ECS poderia realizar a fusão dos sensores). É essa mensagem que permite o funcionamento de nossa arquitetura de simulação. A TAB. 7.2 apresenta a estrutura dessa mensagem. Repare que temos tudo o que precisamos: a atitude, a posição, a velocidade e a velocidade do ar. É por meio dessa mensagem que o piloto automático identifica o estado da aeronave e em seguida realiza o controle necessário.

Campo	Descrição
time_usec	Carimbo do tempo (<i>timestamp</i>)
attitude_quaternion	Atitude representada por quatérnios
rollspeed, pitchspeed, yawspeed	Velocidades angulares do corpo da aeronave [rad/s]
lat, lon	Latitude e longitude * 1E7
alt	Altitude [mm] * 1000
vx, vy, vz	Componentes vetoriais da velocidade inercial [m/s] * 100
ind_airspeed	Velocidade do ar indicada [m/s] * 100
true_airspeed	Velocidade do ar verdadeira [m/s] * 100
xacc, yacc, zacc	Componentes vetoriais da aceleração [mg]

TAB. 7.2: Estrutura da mensagem HIL_STATE_QUATERNION enviada para o piloto automático.

7.2.5 PLUGIN DE FUSÃO SENSORIAL

O plugin de fusão sensorial é a parte central para o cálculo das estimativas dos estados do VANT. Para o seu desenvolvimento utilizou-se a API em C++ (*Application Program Interface*) do X-Plane® responsável por prover uma série de facilidades nas interações com o simulador. A principal delas é a capacidade de extrair dados da cinemática, dinâmica, atmosfera e sensores. Para as operações de suavização incremental, utilizamos a biblioteca GTSAM (*Georgia Tech Smoothing and Mapping*). Outra facilidade da API do X-Plane® é a possibilidade de manipulações gráficas. Com isso, criamos uma pequena caixa no canto superior direito da tela do simulador para apresentar em tempo real as informações necessárias. Isso possibilitou rápidas inspeções sobre a qualidade das estimativas. Nessa caixa são apresentados os dados dos sensores inerciais (acelerômetro e girômetro) e os valores reais, incrementais e os erros para a orientação, posição, velocidade e o vento. A estimativa do vento foi calculada mas não utilizada, uma vez que era possível enviar o dado do sensor de pressão diferencial diretamente, via MAVLink, para o piloto automático. Com relação ao magnetômetro, o X-Plane® somente possuia o valor do ângulo magnético (ϕ_{mag}). Para calcular os campos magnéticos nos eixos x, y e z, utilizou-se o *World Magnetic Model 2010-2015*(MAUS, 2010). O modelo calcula as componentes tridimensionais do campo magnético e a declinação em relação ao norte magnético, mediante a entrada da latitude, da longitude e da altitude. Esse cálculo somente é feito no início da execução do plugin, uma vez que ele pode ser considerado constante para voos com deslocamentos

pequenos. Finalmente, para obter os campos no corpo da aeronave o campo magnético \mathbf{m}^i deve ser rotacionado conforme

$$\mathbf{m}^b = \mathcal{R}_i^b \mathbf{m}^i \quad (7.1)$$

O Algoritmo 6 apresenta o pseudocódigo do plugin de fusão sensorial.

```

Time
  time: 0.000000
Sensors
  Accel: 0.29 m/s2 0.00 m/s2 -10.66 m/s2
  Gyro: -0.06 rad/s; 0.05 rad/s ;-0.07 rad/s
Attitude
  phi: -7.19 theta: 3.39 psi: -75.48
  phi: -7.58 theta: 3.17 psi: -75.02
  phi: 0.39 theta: 0.22 psi: 0.46
Local Position
  x: 9343.153959 y: 34465.983026 z: 71.829623
  x: 9344.494030 y: 34467.554763 z: 72.141823
  x: 1.340071 y: 1.571738 z: 0.312201
Global Position
  lat: -22.915714 lon: -43.163857 alt: 28.13897
  lat: -22.915703 lon: -43.163843 alt: 27.83523
Velocity
  vx: 4.679608 vy: -18.080439 vz: -1.006868
  vx: 4.854847 vy: -18.178363 vz: -1.078402
  vx: 0.175239 vy: 0.097925 vz: 0.071534
Wind
  Windx: 0.171098 Windy: 0.137245 Windz: -0.982
  Windx: 0.000000 Windy: 0.000000 Windz: 0.0000
  Windx: 0.171098 Windy: 0.137245 Windz: 0.9820

```

FIG. 7.6: Tela do plugin de fusão sensorial.

Algoritmo 6: Plugin de fusão sensorial. Obtém os dados dos sensores do simulador e calcula as estimativas incrementais para os estados do VANT.

Entrada: Dados dos sensores: acelerômetro, girômetro, barômetro, GPS e pressão diferencial.

Saída: Estimativas dos estados do VANT: $\{p_n, p_e, p_d, v_n, v_e, v_d, \phi, \theta, \psi\}$ e pressão diferencial}.

`dadosMagnetometro ← SimularDadosMagnetometro(latitude, longitude, altitude,
modeloMagnetico);`

Enquanto simulação estiver rodando **faça**

```
dadosSensores ← ExtrairDadosSensoresXPlane();  
dadosSensores ← dadosSensores + dadosMagnetometro;  
AdicionarRuidoBranco(dadosSensores);  
pacoteMAVLink ← GerarPacoteMAVLink(dadosSensores);  
EnviarPacoteMAVLink(pacoteMAVLink);  
ReceberDadosControle();
```

fim-enquanto

8 EXPERIMENTOS E RESULTADOS

Neste capítulo, descrevemos as simulações e apresentamos as validações a respeito do algoritmo de suavização incremental. Para esse objetivo, criamos três missões e um ensaio de voo a serem cumpridos pela aeronave: um voo quadrado, um voo montanha-russa e um voo lawn-mower, útil em tarefas de busca e varredura de áreas. Nosso objetivo com esses voos foi estimular diferentes tipos de movimentos. Todos os experimentos usaram as mesmas configurações de ruído representadas na TAB. 8.1.

Sensor	Taxa (Hz)	Ruído (desvio padrão)
Acelerômetro	100	4mg
Girômetro	100	0,05deg/s
Magnetômetro	100	200nT
Barômetro	2	0,5m
GPS	5	2,0; 2,0; 4,0m (N,E,D)

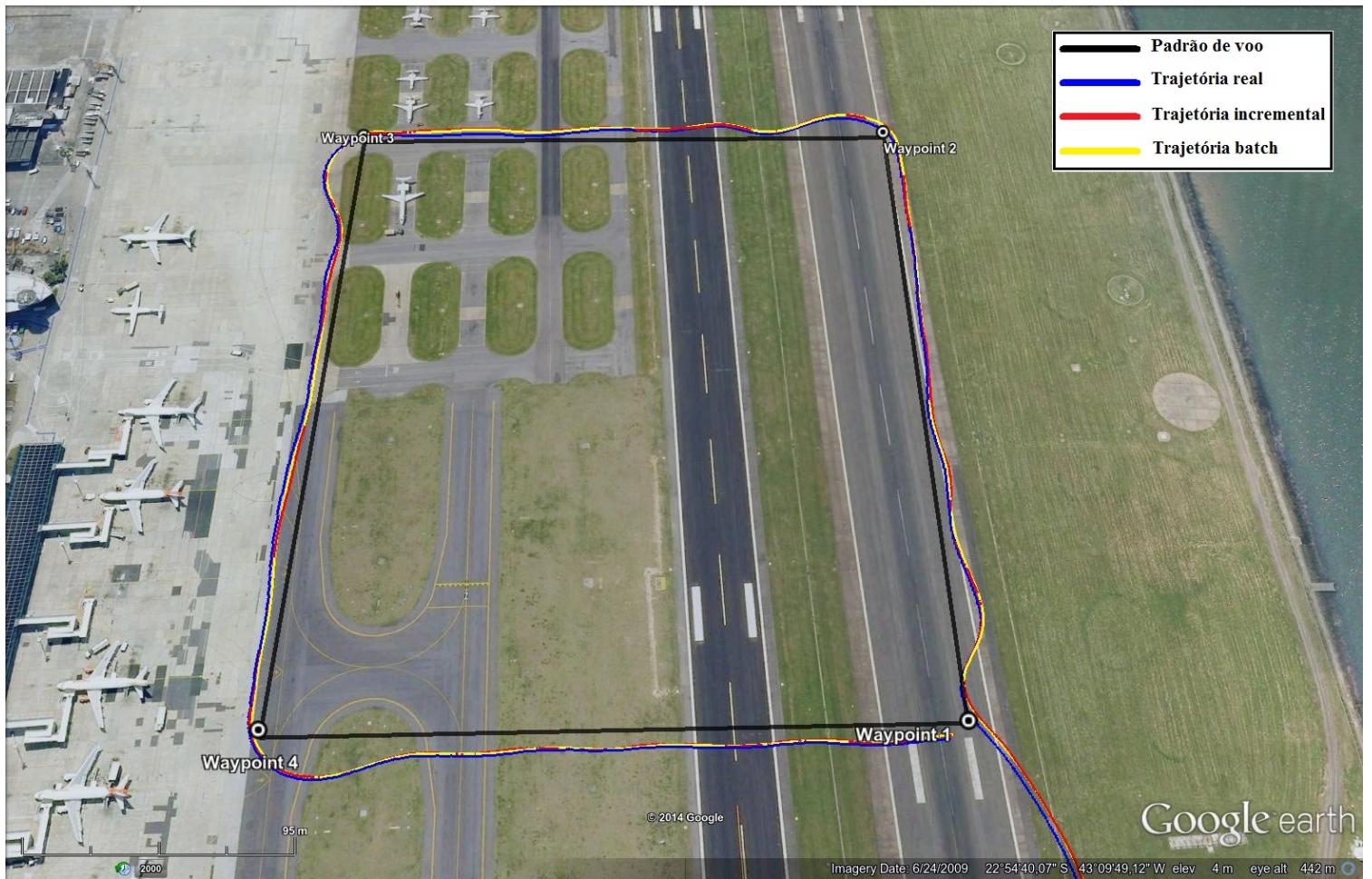
TAB. 8.1: Taxas de operação e ruídos dos sensores utilizados durante os experimentos.

8.1 EXPERIMENTOS

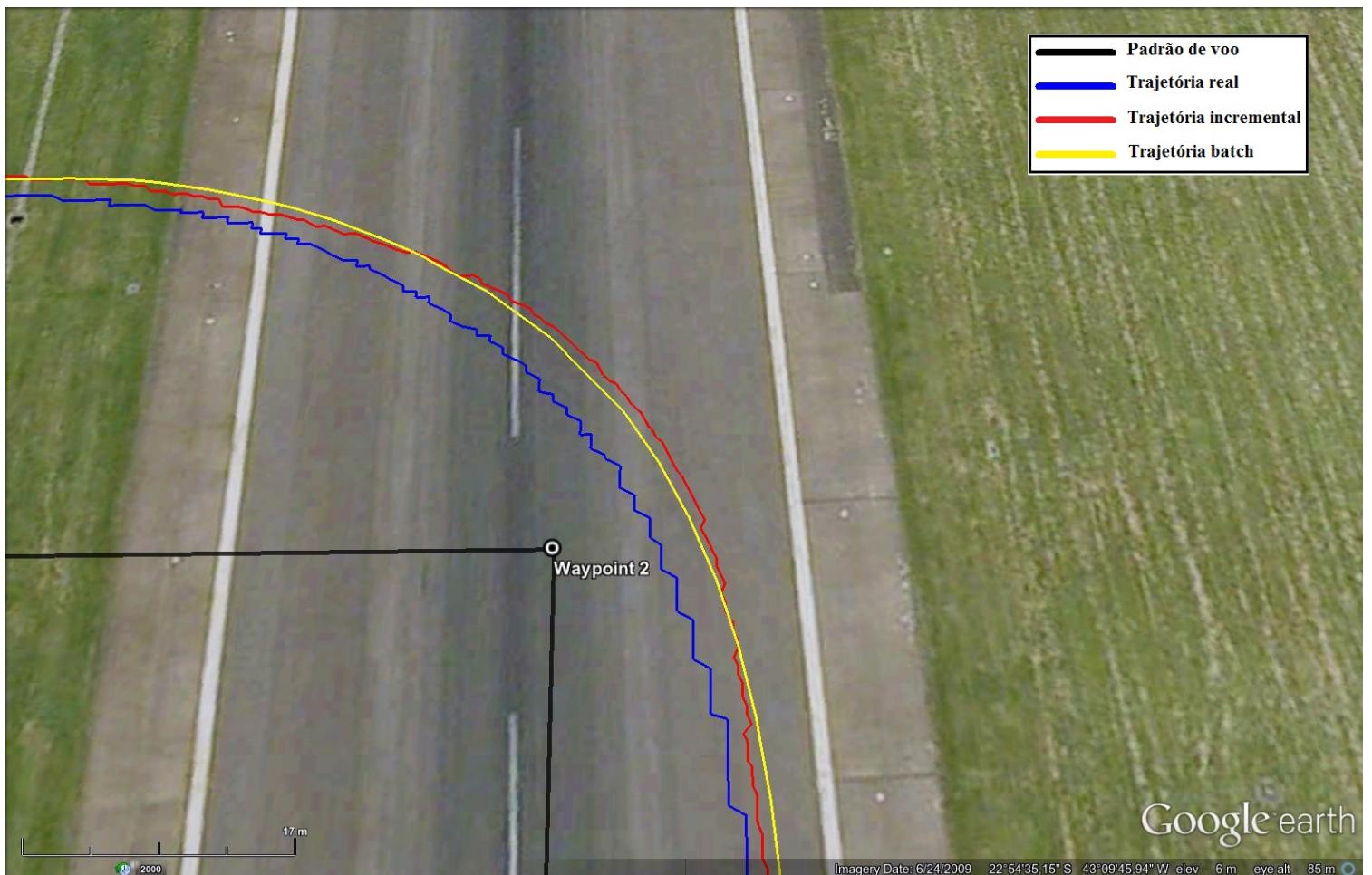
8.1.1 VOO QUADRADO

Neste experimento, a missão a ser cumprida foi um voo quadrado com altitude constante. Como esse era um voo simples, o objetivo principal foi detectar erros no desenvolvimento do plugin e na comunicação entre os componentes. Um problema detectado foi a baixa taxa de atualização causada pelo uso da técnica de pré-integração que diminui a quantidade de nós adicionados ao grafo. A solução foi calcular as previsões realizadas pela pré-integração. Os detalhes foram apresentados na Seção 6.1. O experimento obteve resultados satisfatórios e o voo foi realizado com sucesso. O tempo total de voo foi de 2m:49s. A FIG. 8.1a apresenta o padrão de voo realizado. É possível verificar que as trajetórias se confundem quando estamos mais afastadas. A FIG. 8.1b apresenta uma região aproximada, próximo ao waypoint 2, quando a aeronave realiza um movimento para a esquerda. Não é possível acompanhar perfeitamente o quadrado, pois, fisicamente uma

aeronave de asa fixa não realiza movimentos em "L". Isso explica a trajetória arredonda de 1/4 de círculo da figura aproximada. Outra característica que podemos observar é a aproximação entre as trajetórias real, batch e incremental. Para esse voo o erro absoluto total horizontal (componentes norte e leste) foi de 1,5263m contra 1,4352m da trajetória batch. Além disso, vemos também a característica mais suavizada da trajetória batch (amarelo), onde quase não se vê descontinuidades.



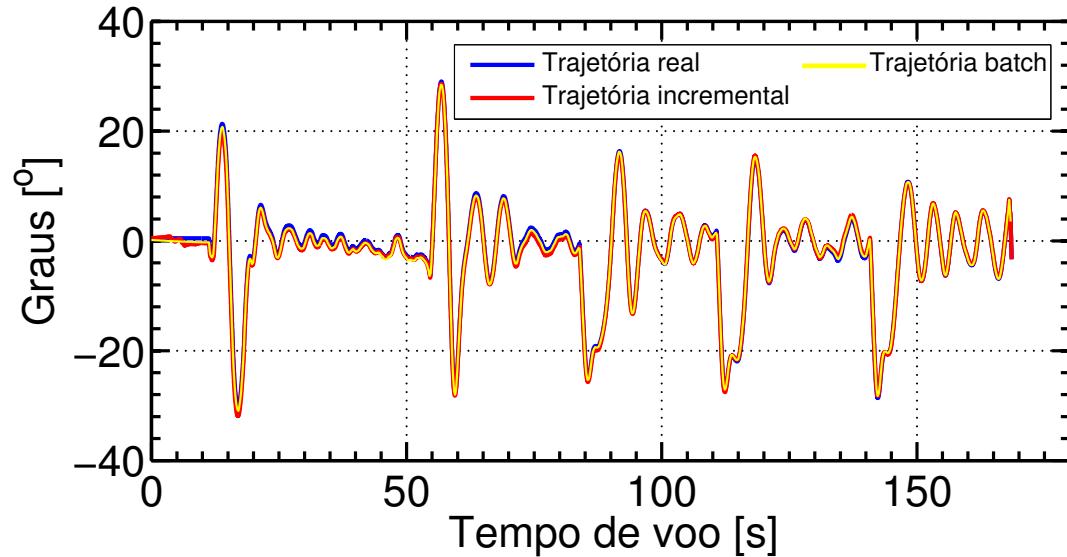
(a) Padrão quadrado (preto).



(b) Região aproximada. A trajetória batch (amarelo) é a mais suavizada.

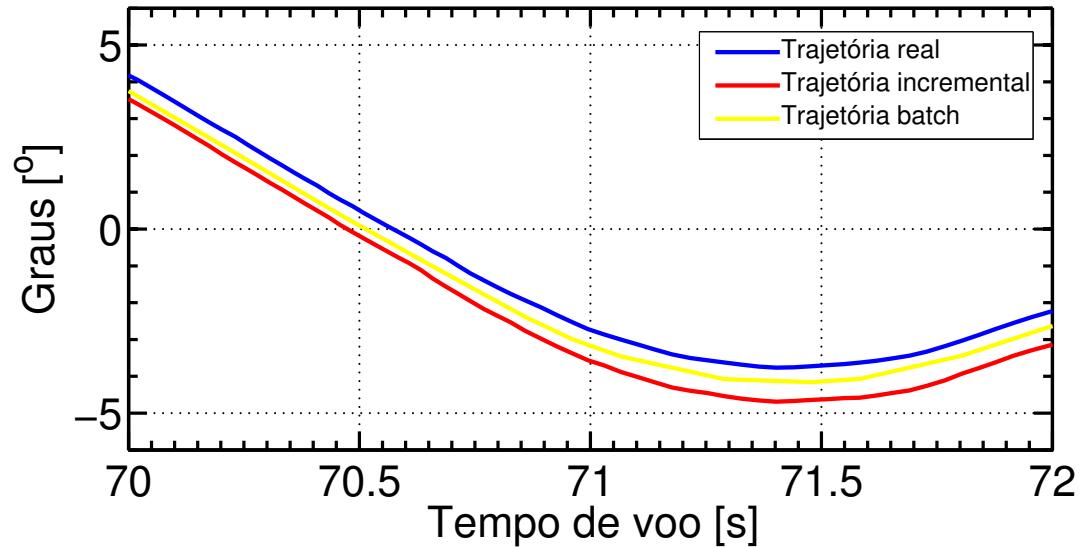
FIG. 8.1: Padrão quadrado (preto) e as trajetórias real (azul), incremental (vermelho) e batch (amarelo).

Rolamento (ϕ)



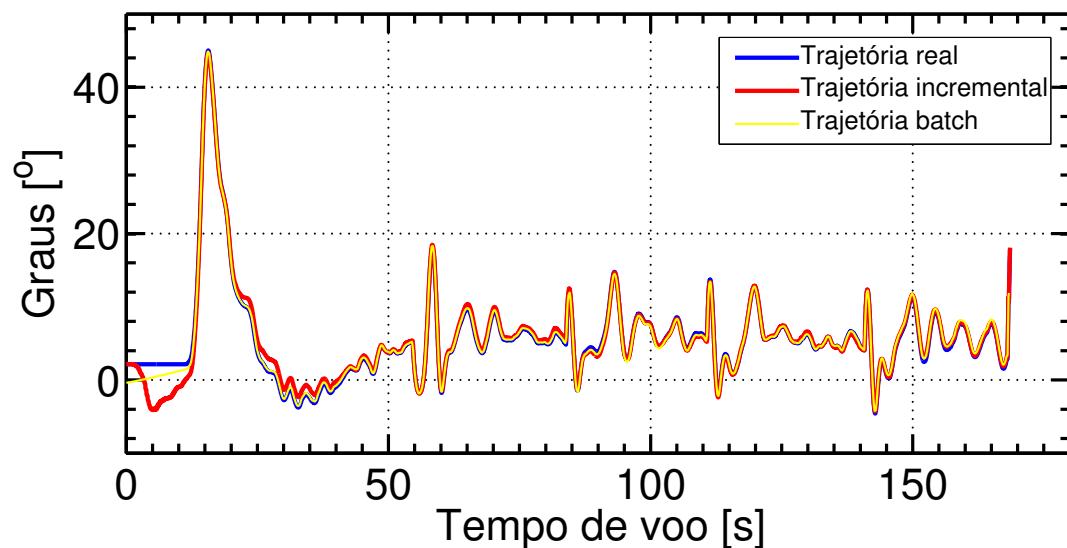
(a) Estimativa para o rolamento.

Rolamento (ϕ) – região aproximada



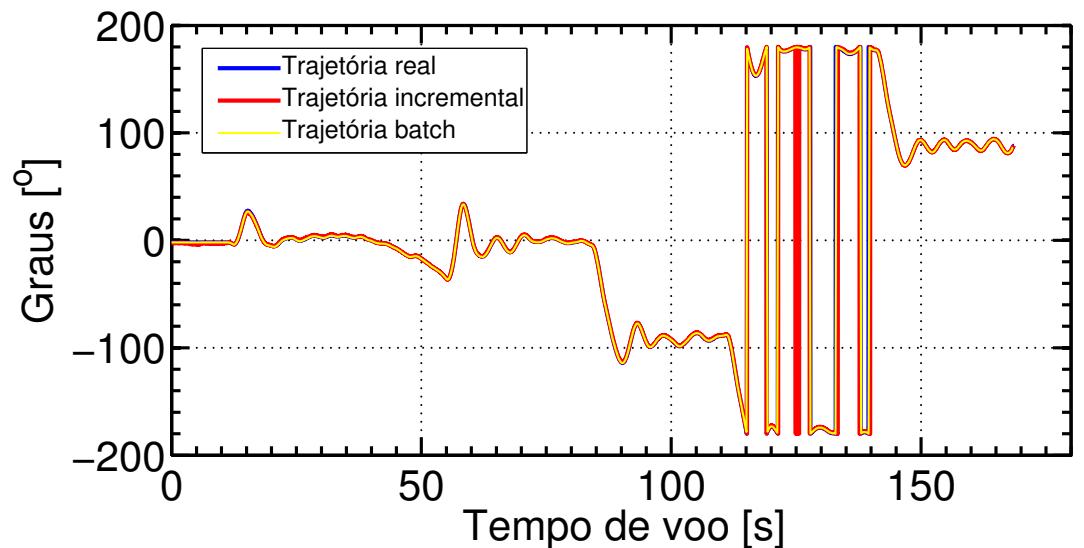
(b) Região aproximada para o rolamento.

Arfagem (θ)



(c) Estimativa para a arfagem.

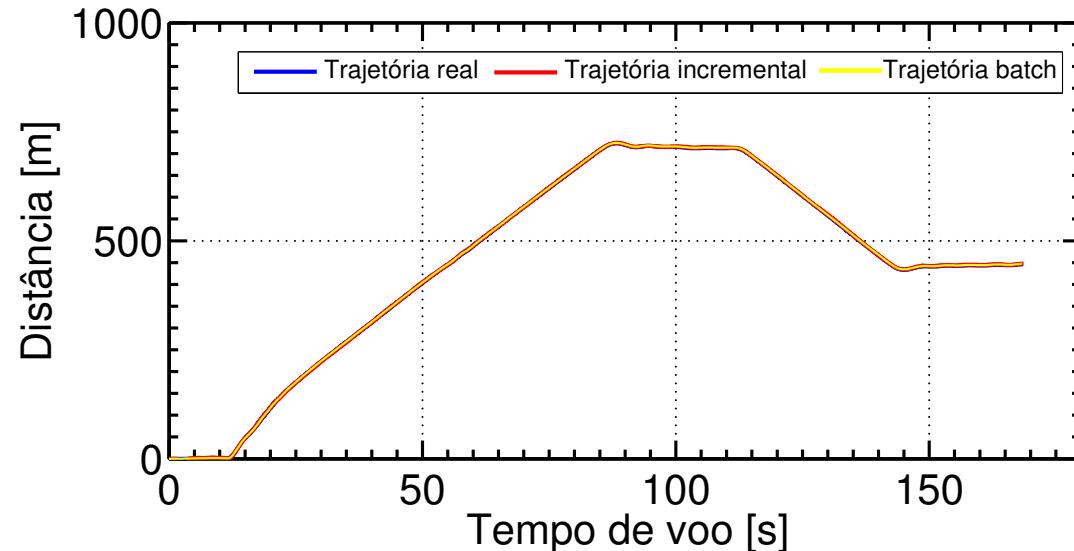
Guinada (ψ)



(d) Estimativa para a guinada.

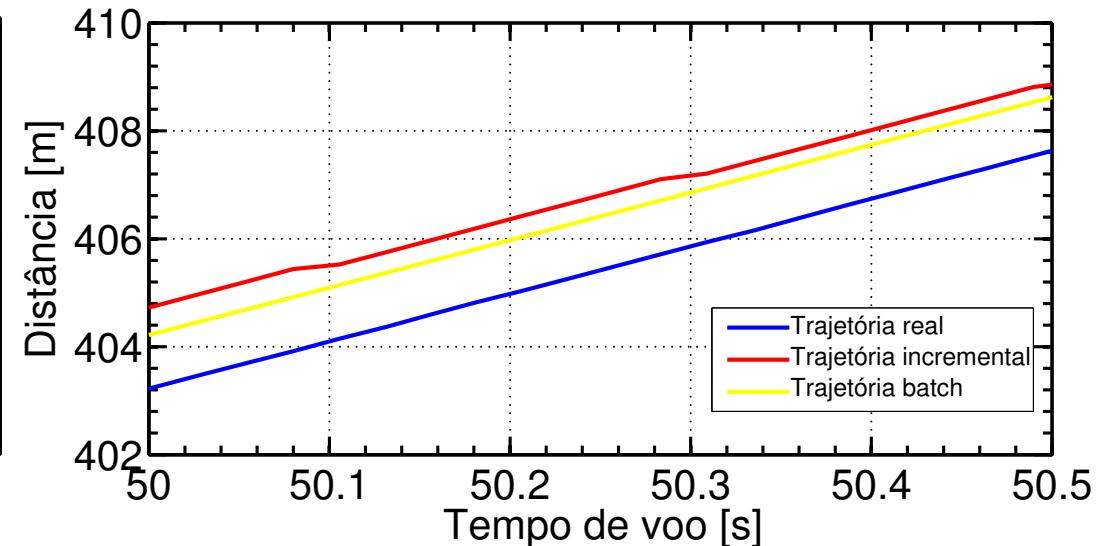
FIG. 8.2: Voo quadrado. Estimativas para a orientação.

Posição Norte (x)



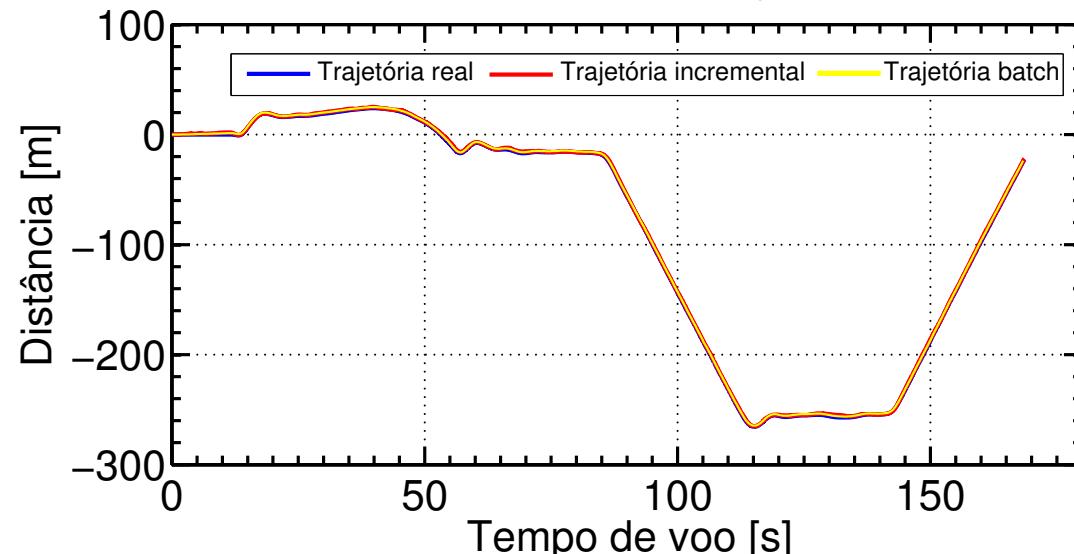
(a) Estimativa para a posição norte.

Posição Norte (x) – região aproximada



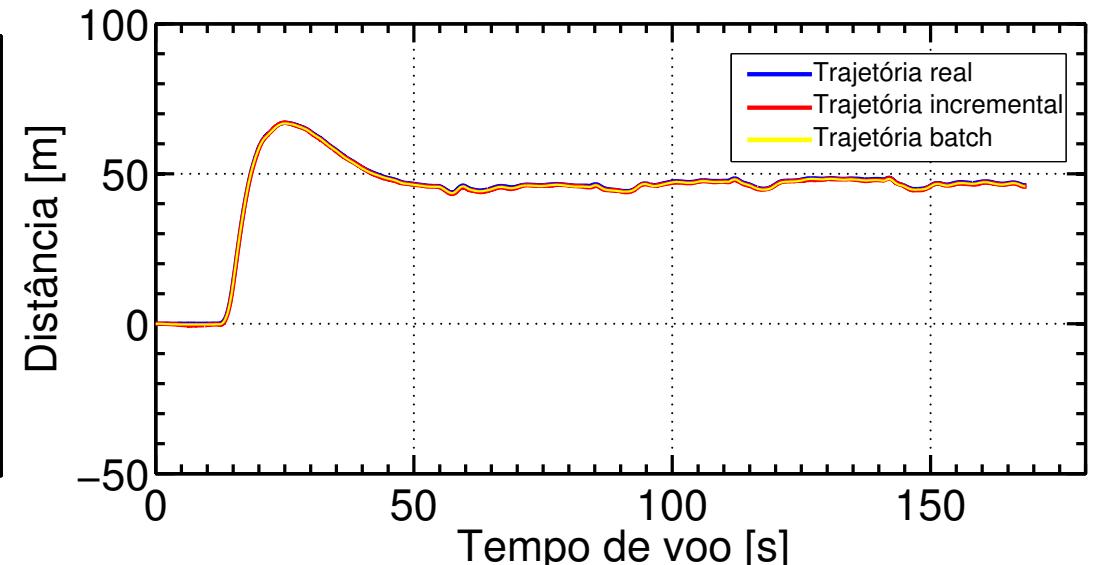
(b) Região aproximada para a posição norte.

Posição Leste (y)



(c) Estimativa para a posição leste.

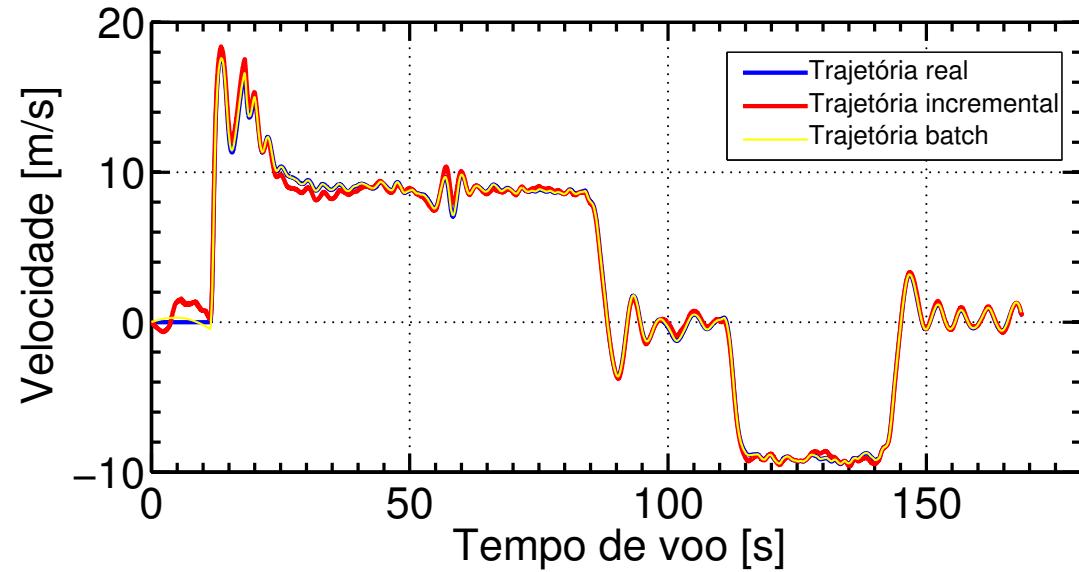
Posição Abaixo (z)



(d) Estimativa para a posição abaixo.

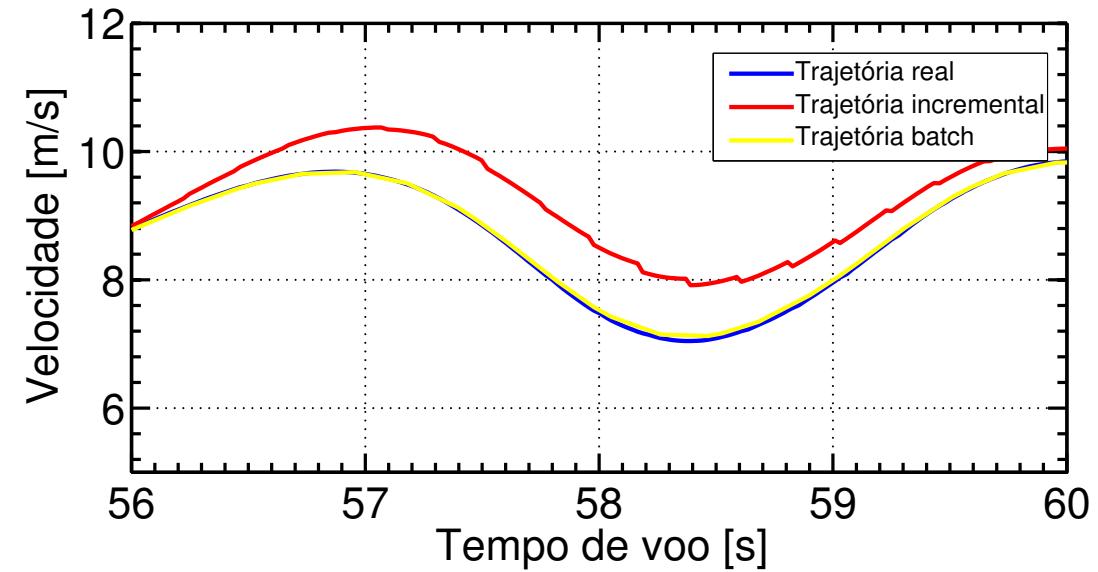
FIG. 8.3: Voo quadrado. Estimativas para a posição.

Velocidade Norte (x)



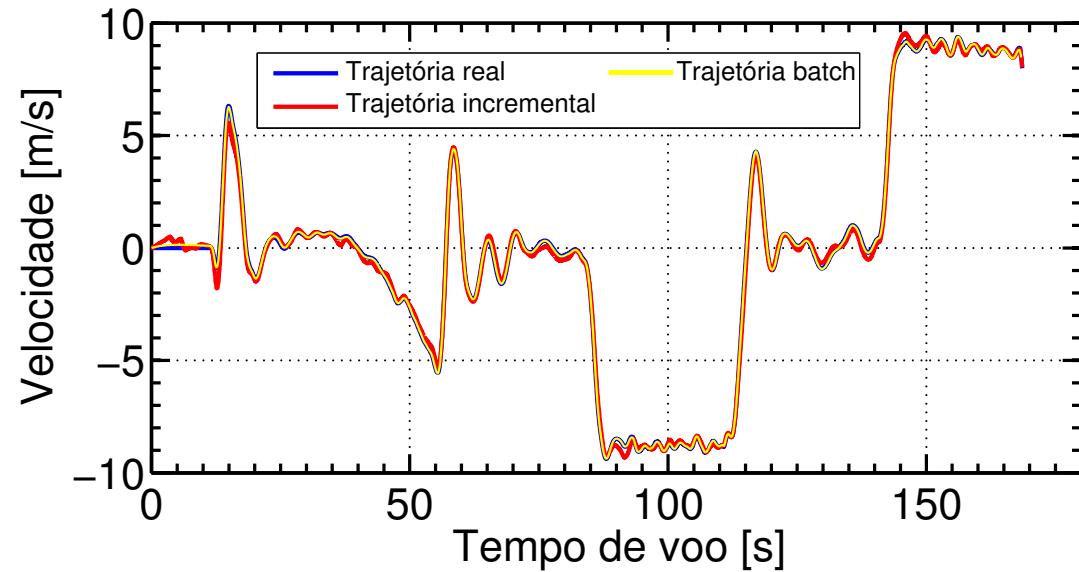
(a) Estimativa para a velocidade norte.

Velocidade Norte (x) – região aproximada



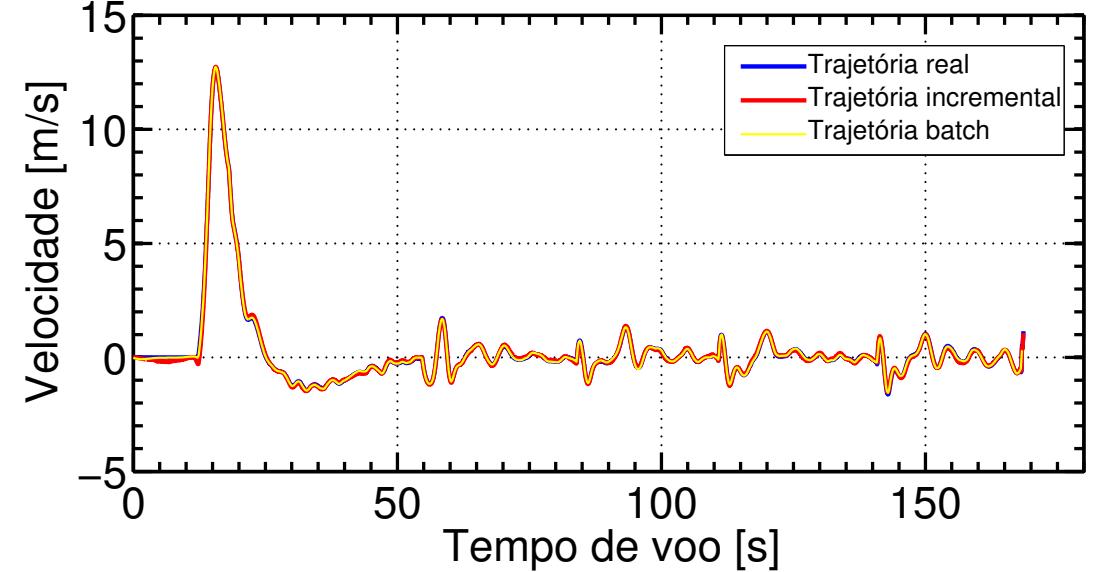
(b) Região aproximada para a velocidade norte.

Velocidade Leste (y)



(c) Estimativa para a velocidade leste.

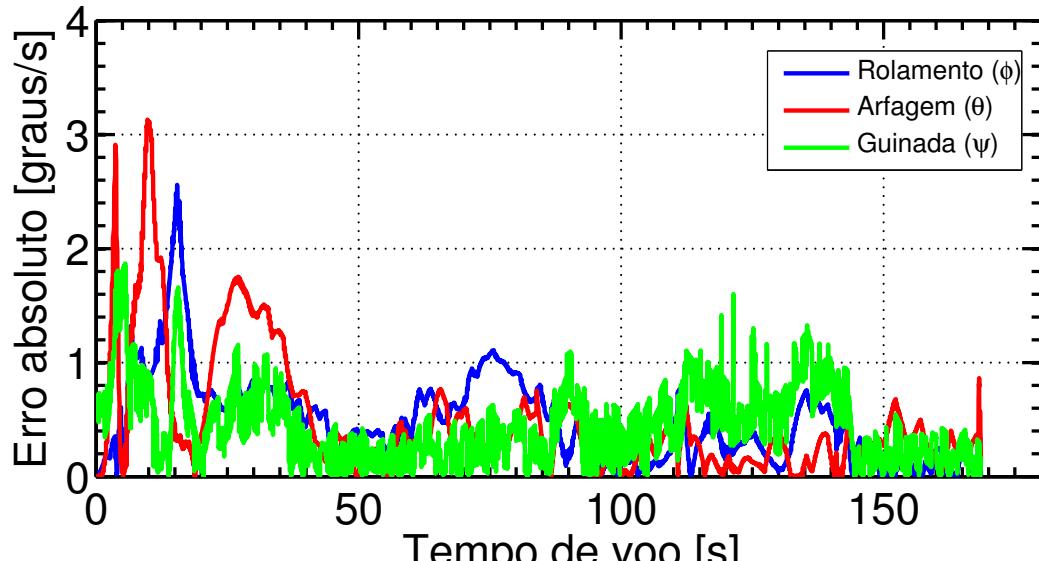
Velocidade Abaixo (z)



(d) Estimativa para a velocidade abaixo.

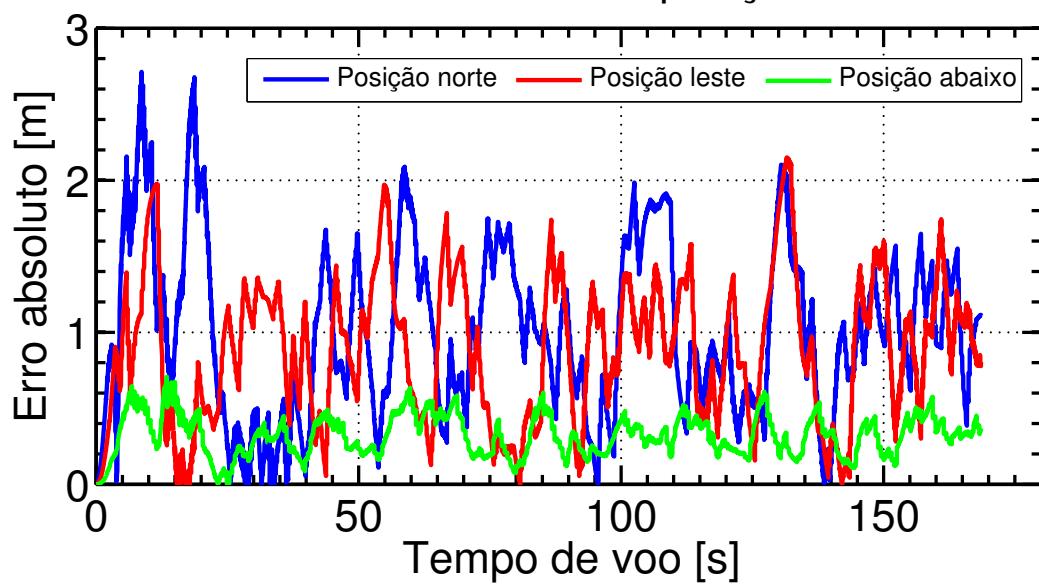
FIG. 8.4: Voo quadrado. Estimativas para a velocidade.

Erro absoluto da orientação



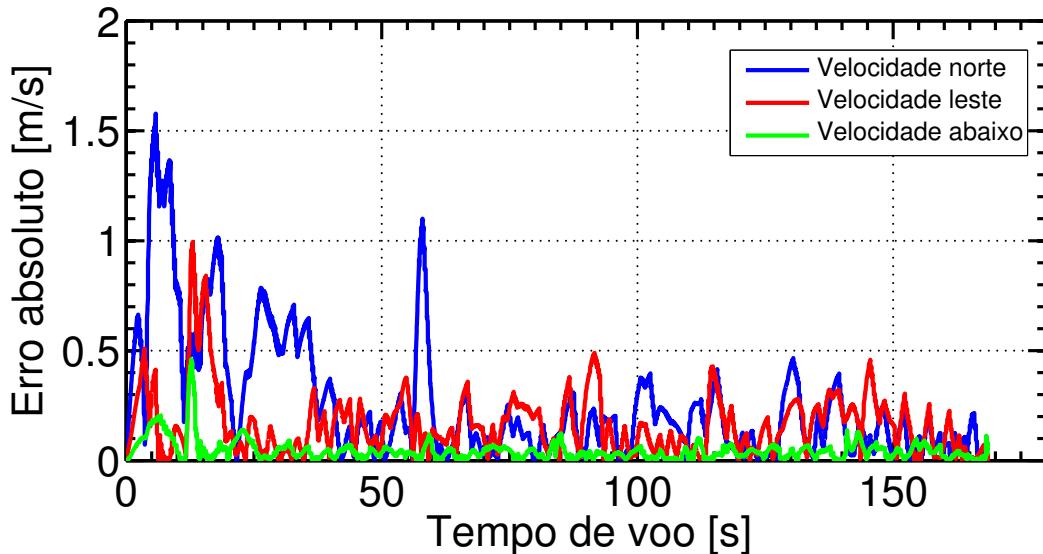
(a) Erro absoluto da orientação.

Erro absoluto das posições



(b) Erro absoluto da posição.

Erro absoluto das velocidades

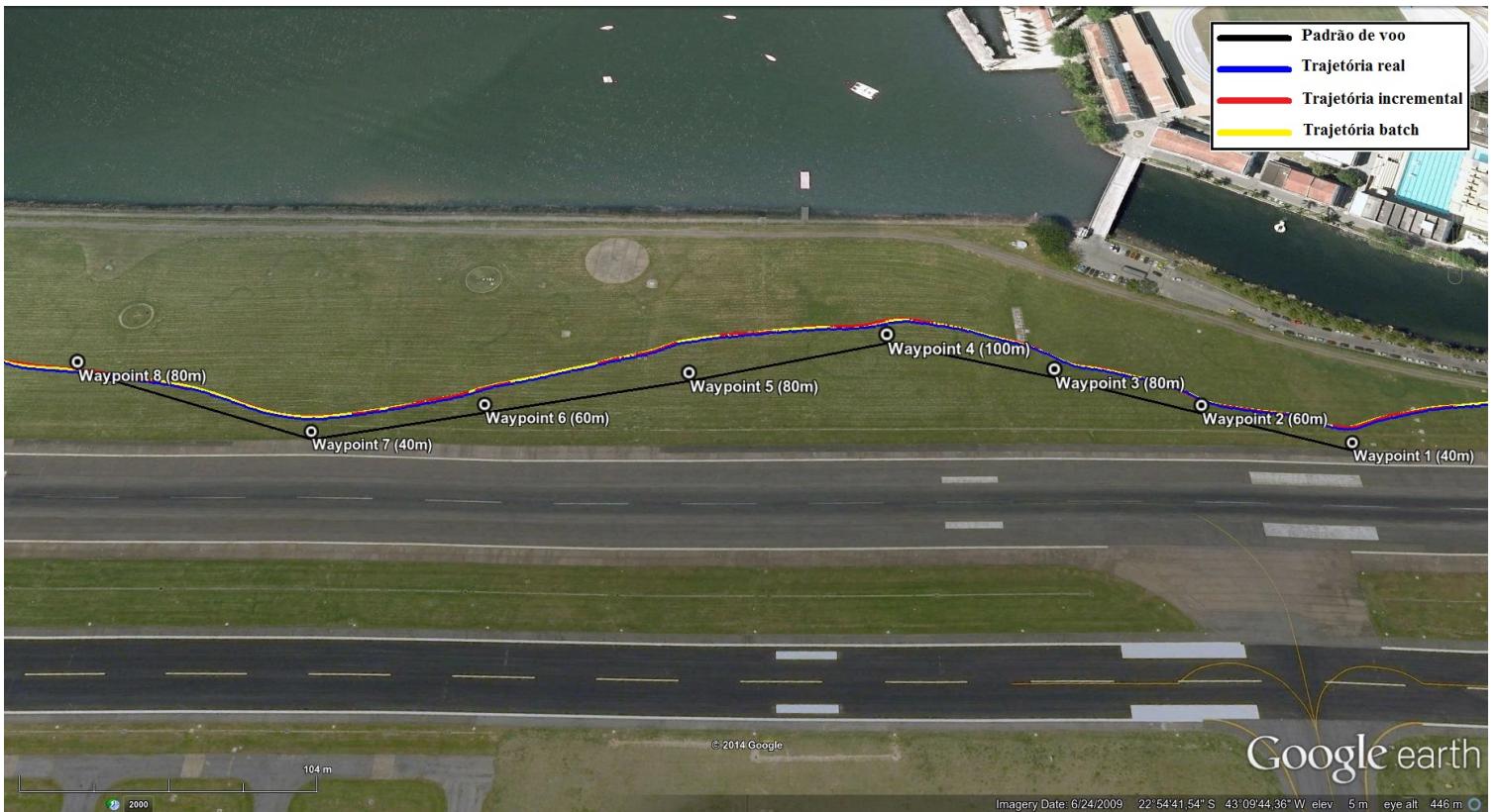


(c) Erro absoluto da velocidade.

FIG. 8.5: Voo quadrado. Erros absolutos da orientação, posição e velocidade.

8.1.2 VOO MONTANHA-RUSSA

O propósito deste experimento foi validar a estimativa da altitude, uma vez que nos outros experimentos a consideramos constante. Para esse objetivo, nós criamos um voo que denominamos de montanha-russa, onde a aeronave sobe e desce. A missão seguiu o seguinte padrão de altitudes: 40-60-80-100-80-60-40-80m. Como esperado, esse voo apresentou a pior estimativa para a arfagem, o movimento mais afetado pelos movimentos de subida e descida. O erro médio quadrático foi de 1,415m, contra os valores de 1,115m e 0,786m obtidos nos outros experimentos. Em relação à estimativa da altitude, apesar do pior resultado, verificamos que os erros médios ficaram praticamente constantes nos três experimentos (0,389m contra 0,351m e 0,379m). Esse fato pode ser explicado pela influência do barômetro, que proporciona informações mais precisas em relação à altitude. A FIG. 8.6a ilustra o voo, que durou 1m:30s e a FIG. 8.6b mostra uma região aproximada selecionada.



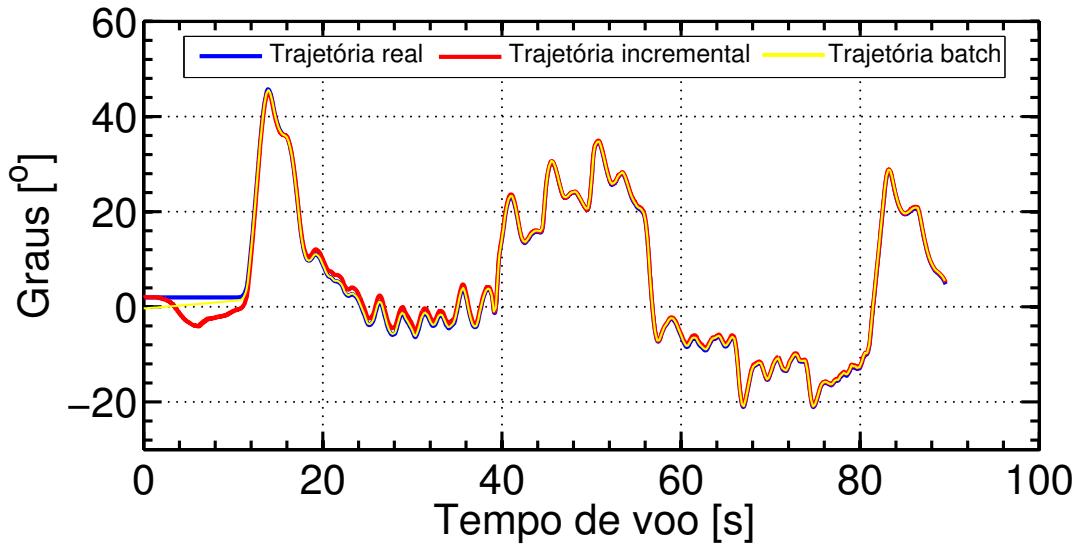
(a) Padrão montanha-russa (preto).



(b) Região aproximada. A trajetória batch (amarelo) é a mais suavizada.

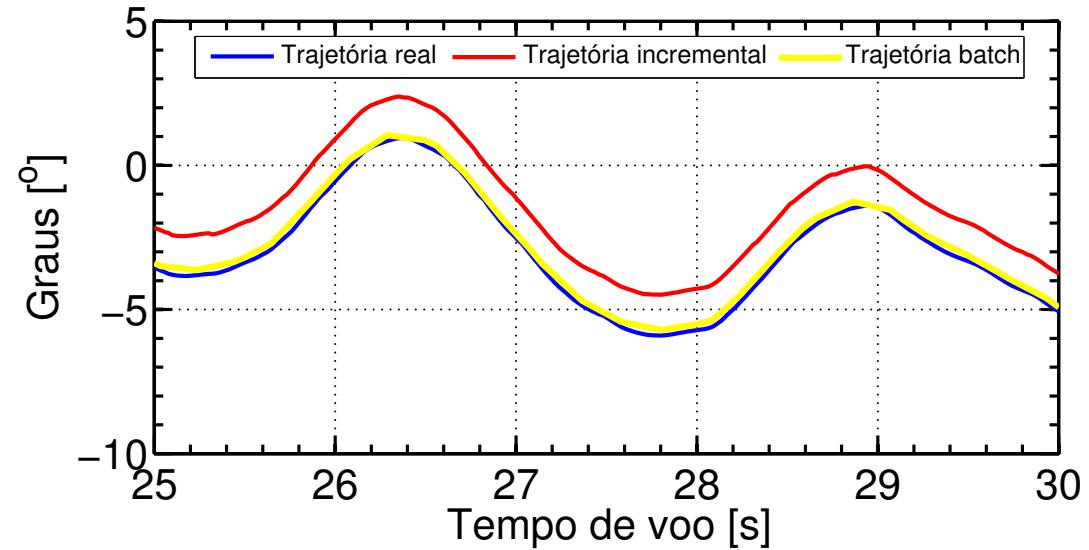
FIG. 8.6: Padrão montanha-russa (preto) e as trajetórias real (azul), incremental (vermelho) e batch (amarelo).

Arfagem (θ)



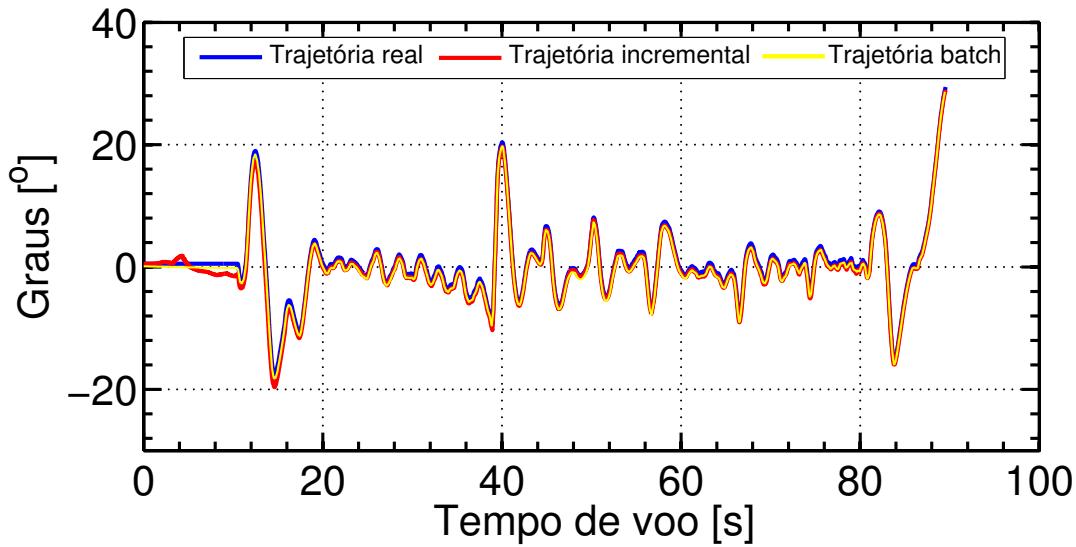
(a) Estimativa para o arfagem.

Arfagem (θ) – região aproximada



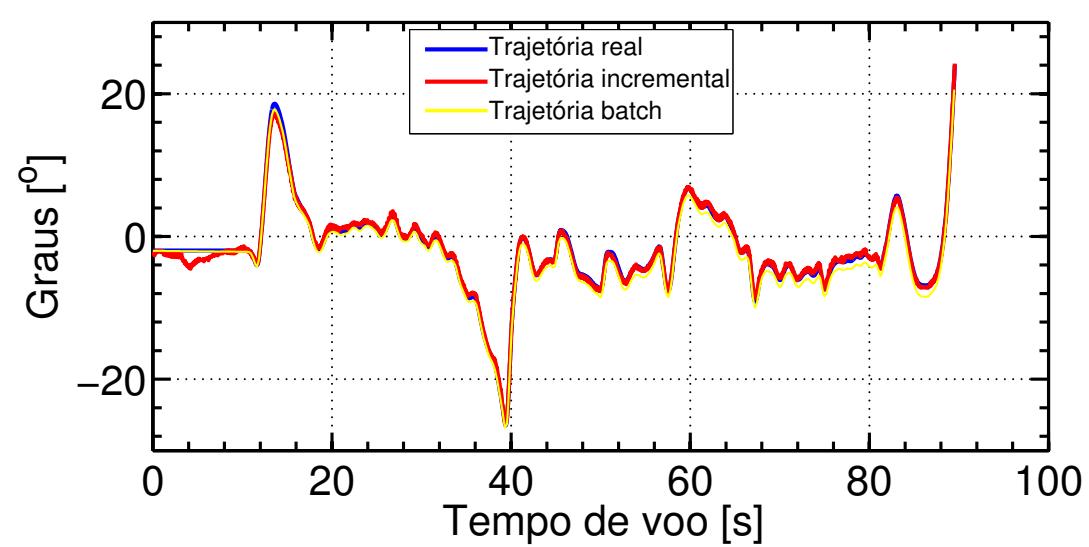
(b) Região aproximada para a arfagem.

Rolamento (ϕ)



(c) Estimativa para o rolamento.

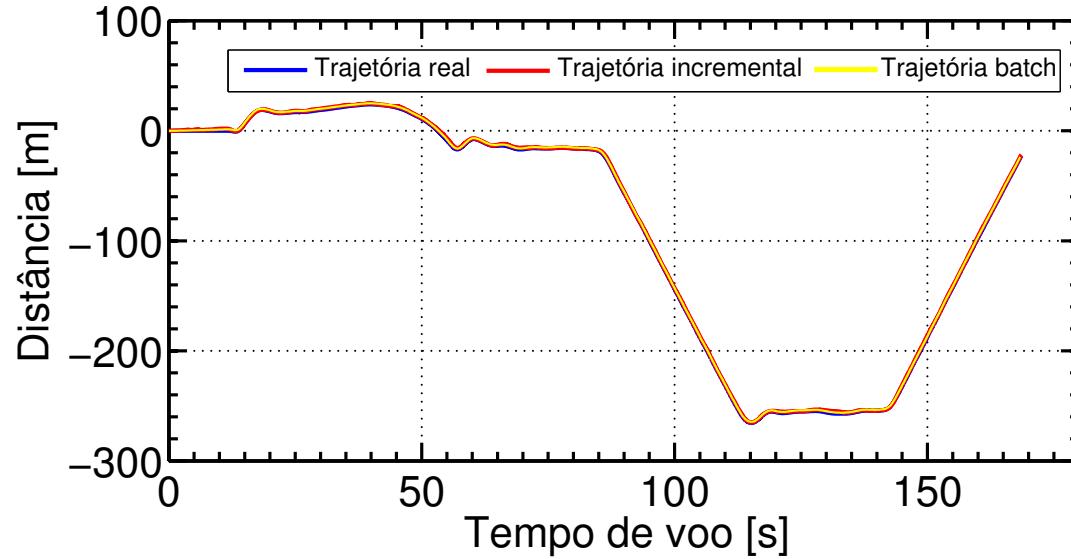
Guinada (ψ)



(d) Estimativa para a guinada.

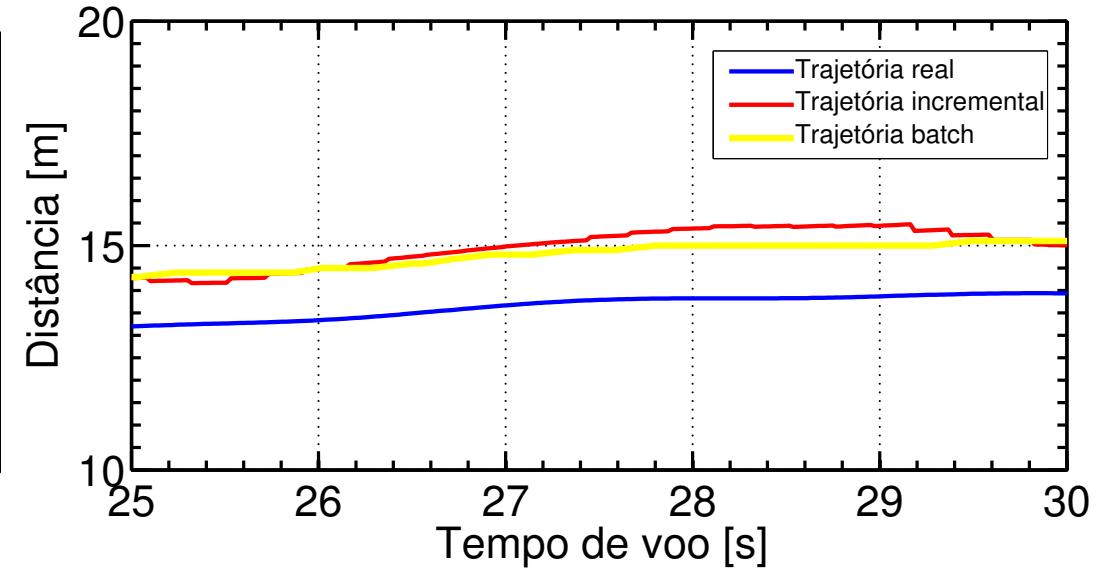
FIG. 8.7: Voo montanha russa. Estimativas para a orientação.

Posição Leste (y)



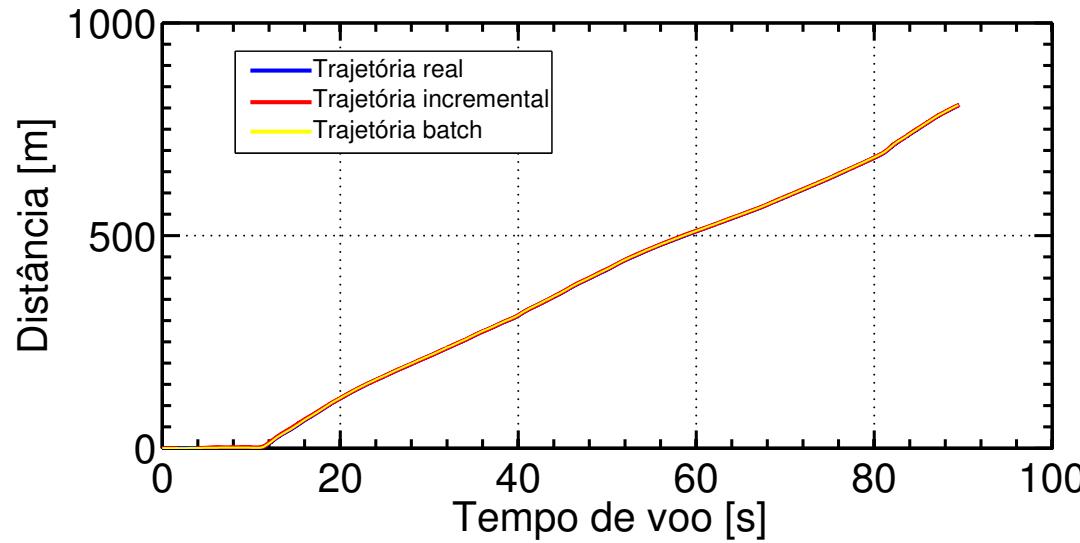
(a) Estimativa para a posição leste.

Posição Leste (y) – região aproximada



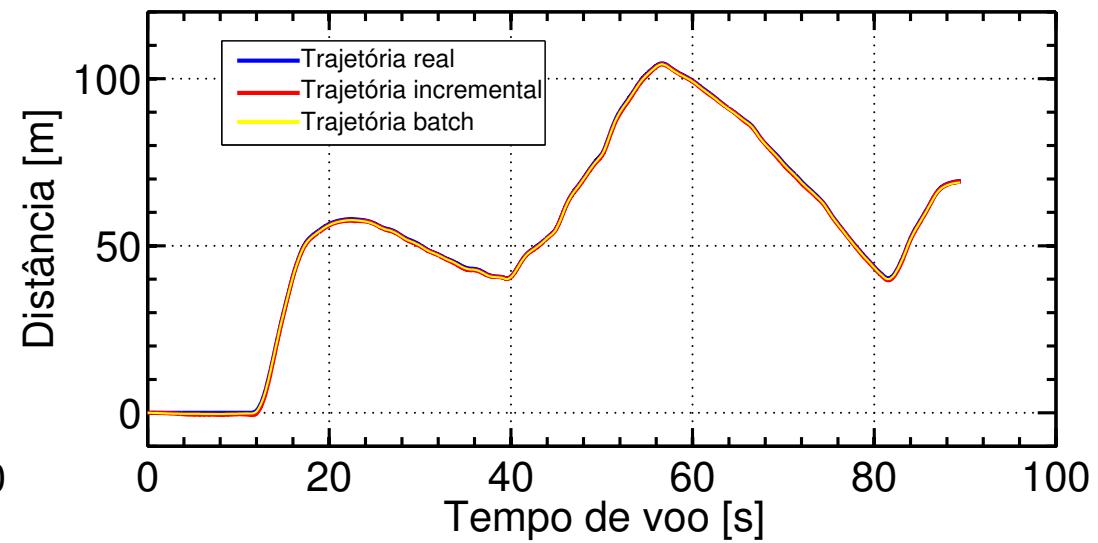
(b) Região aproximada para a posição leste.

Posição Norte (x)



(c) Estimativa para a posição norte.

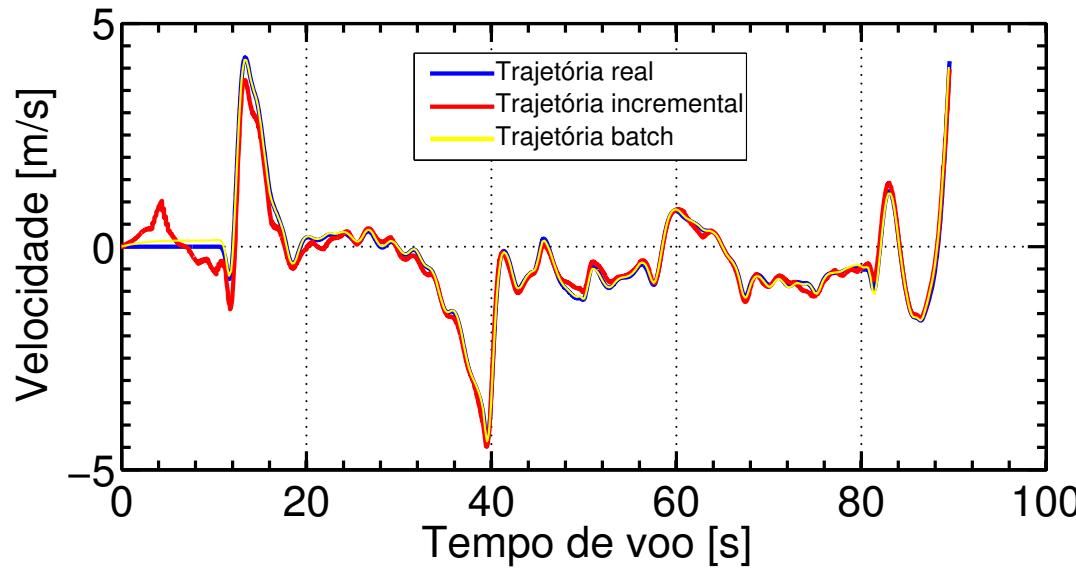
Posição Abaixo (z)



(d) Estimativa para a posição abaixo.

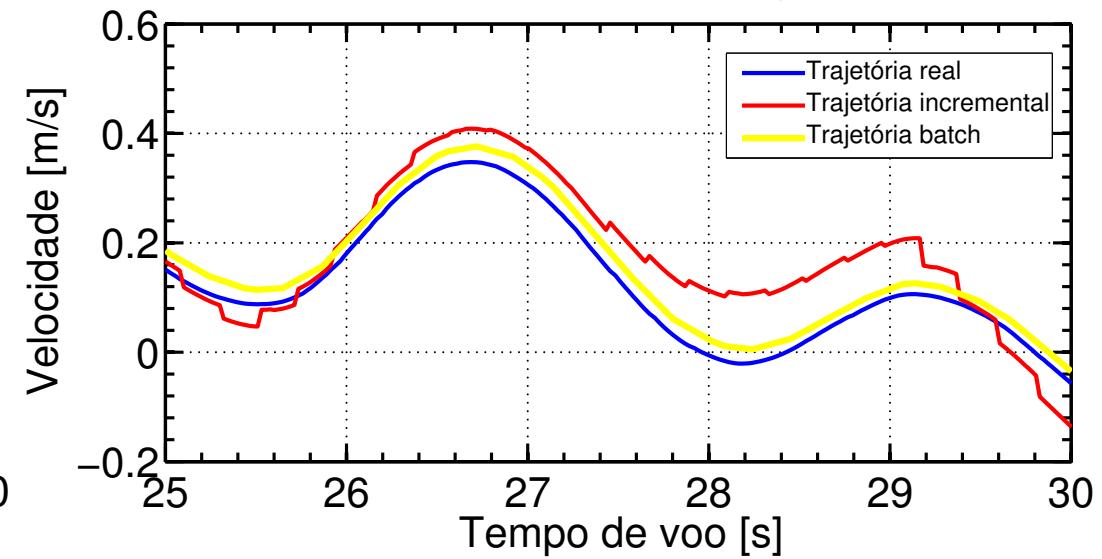
FIG. 8.8: Voo montanha russa. Estimativas para a posição.

Velocidade Leste (y)



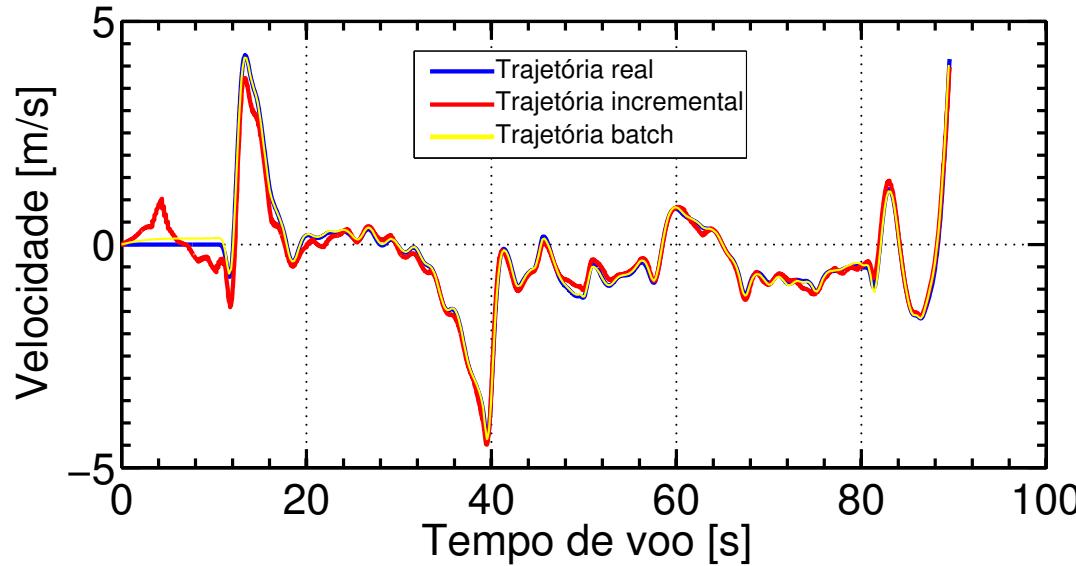
(a) Estimativa para a velocidade leste.

Velocidade Leste (y)



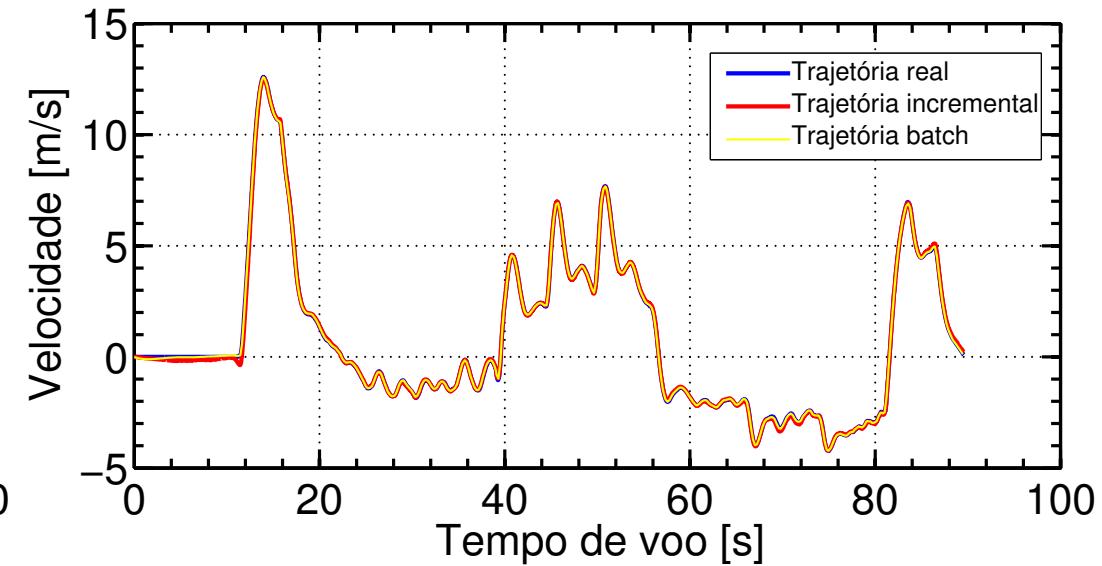
(b) Região aproximada para a velocidade leste.

Velocidade Leste (y)



(c) Estimativa para a velocidade norte.

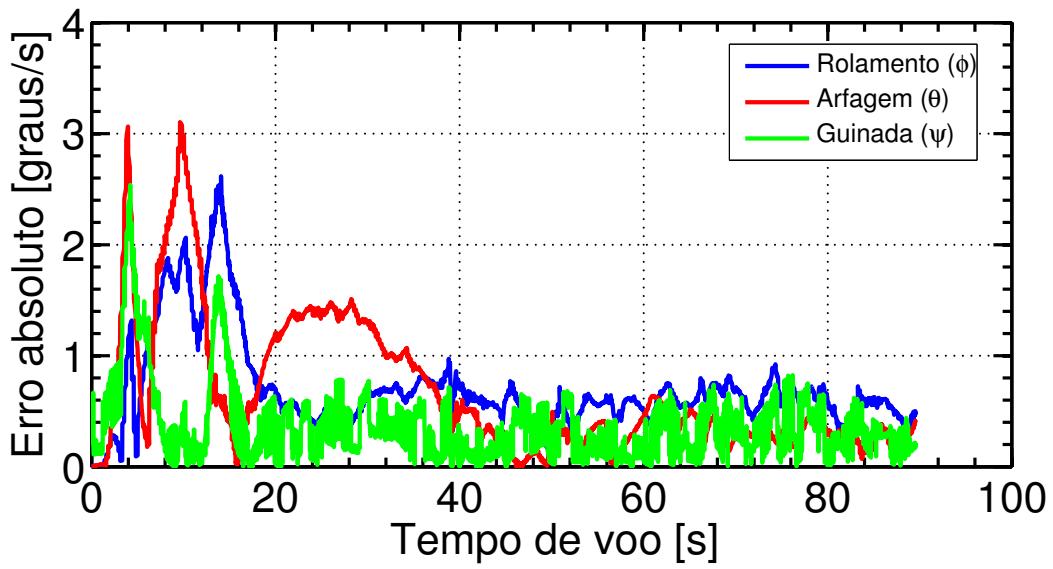
Velocidade Abaixo (z)



(d) Estimativa para a velocidade abixo.

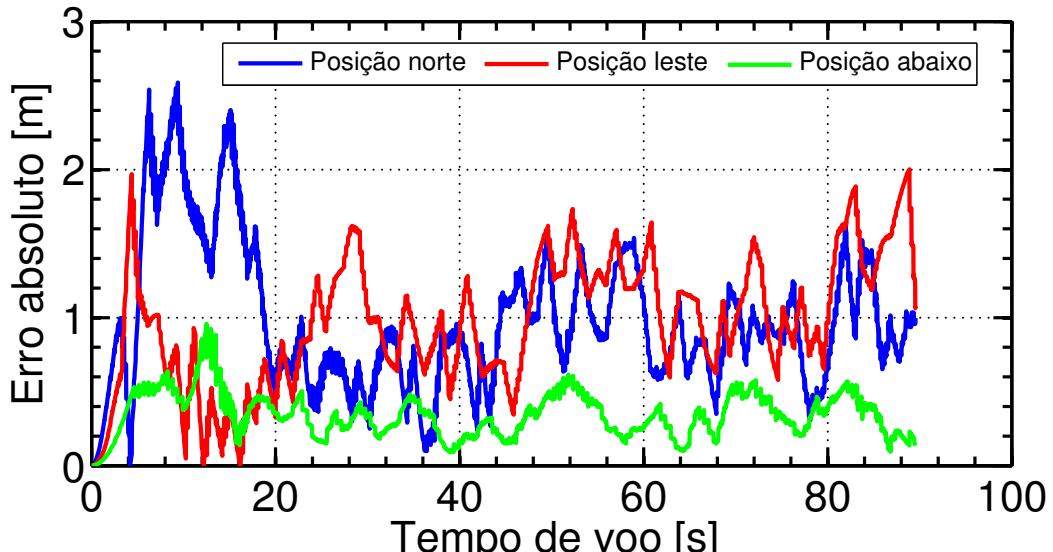
FIG. 8.9: Voo montanha russa. Estimativas para a velocidade.

Erro absoluto da orientação



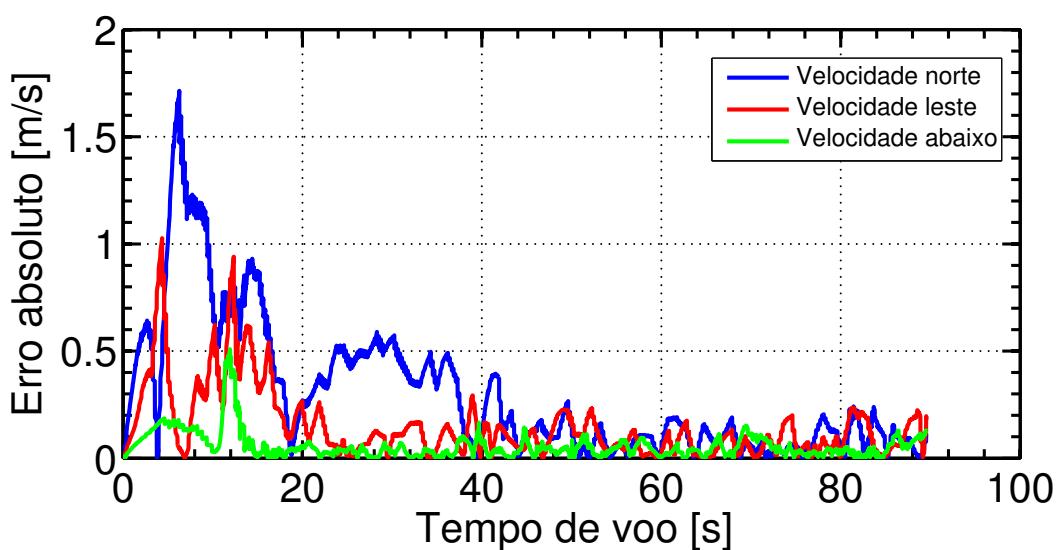
(a) Erro absoluto da orientação.

Erro absoluto das posições



(b) Erro absoluto da posição.

Erro absoluto das velocidades

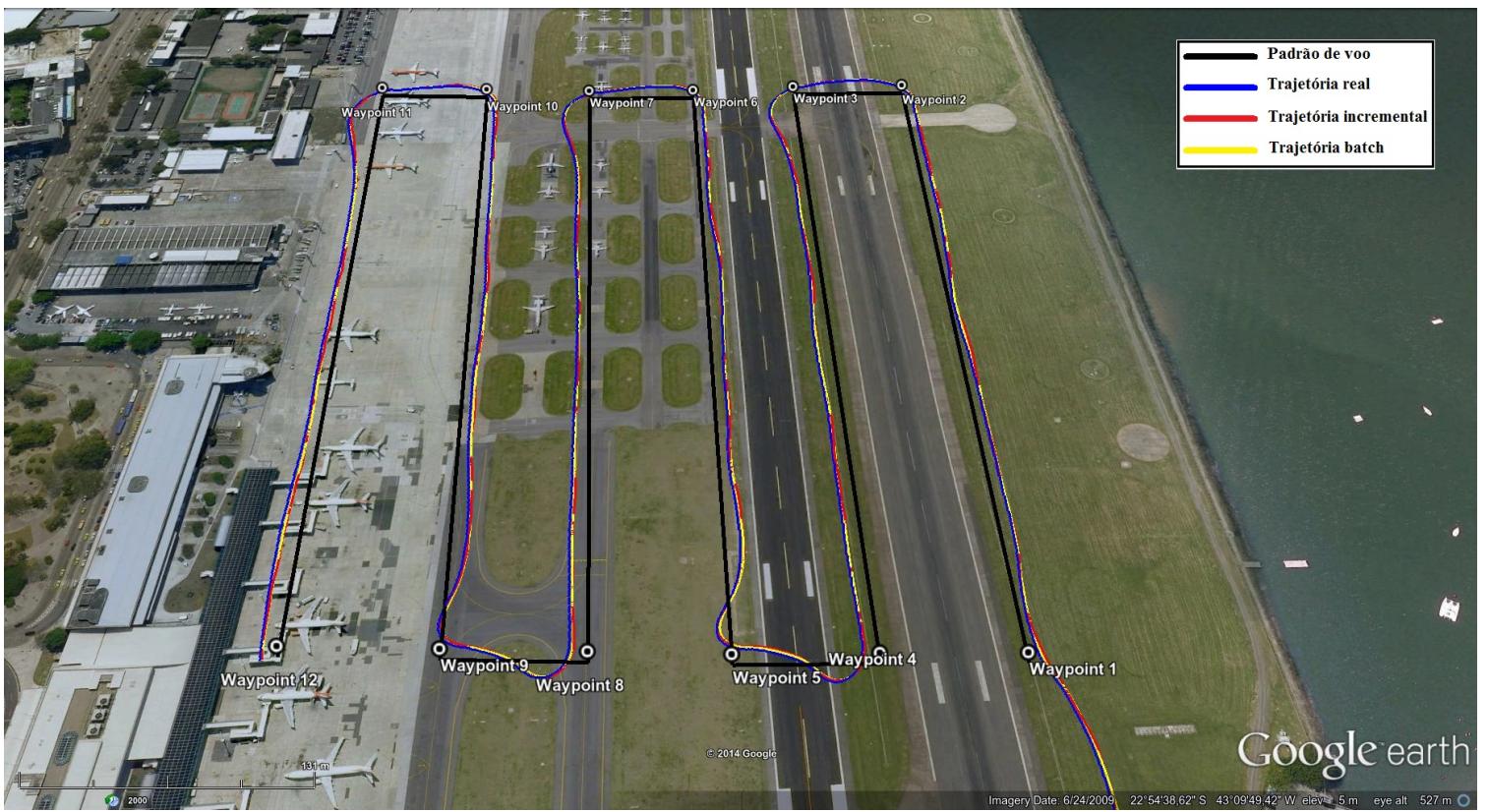


(c) Erro absoluto da velocidade.

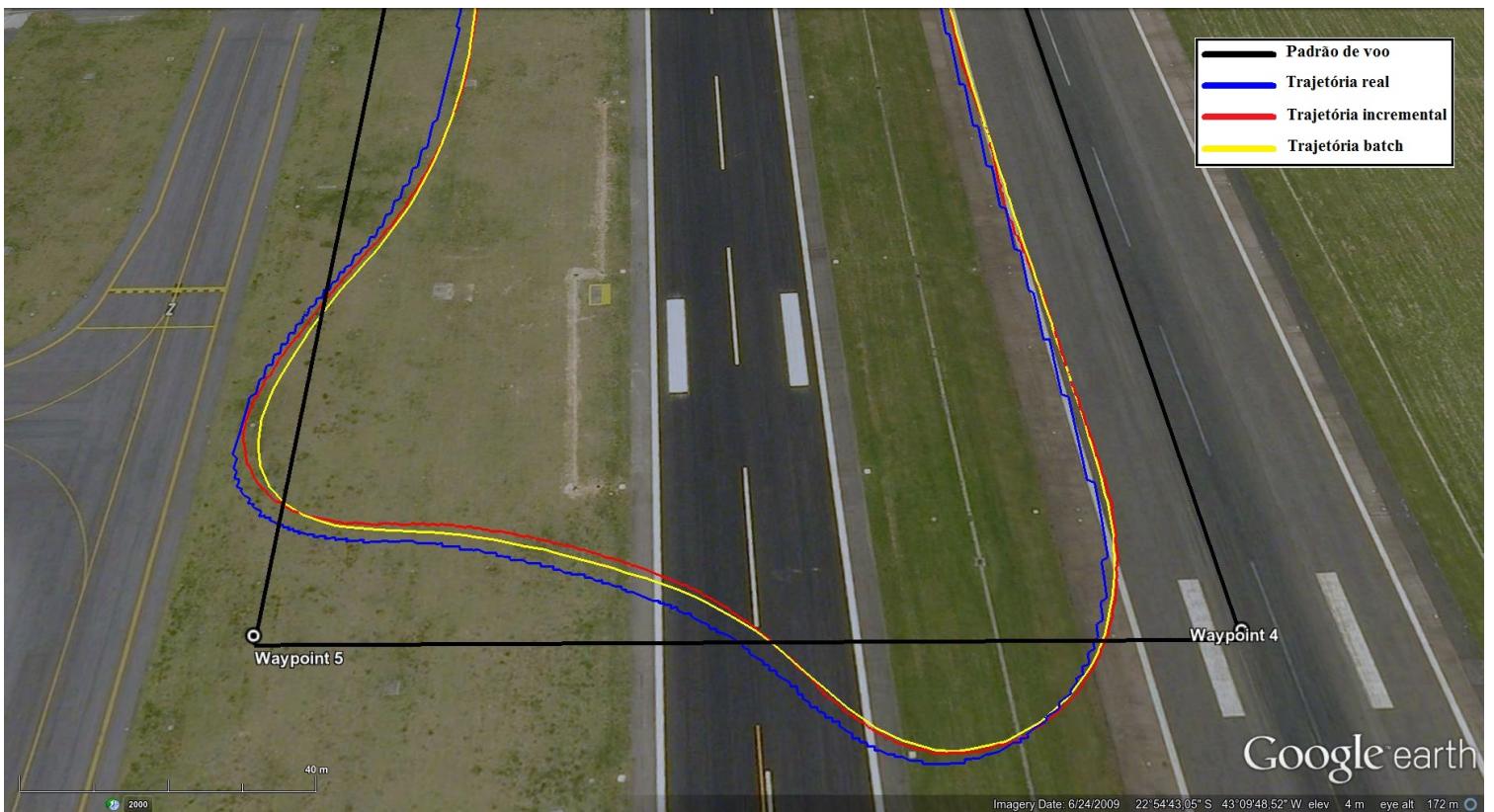
FIG. 8.10: Voo montanha russa. Erros absolutos da orientação, posição e velocidade.

8.1.3 VOO *LAWN-MOWER*

O objetivo deste experimento foi a realização de um padrão de voo que representasse uma missão real e observar como a aeronave reagiria. Para isso utilizamos um dos padrões mais utilizados para busca e monitoramento de áreas: o padrão lawn-mower (OUSING-SAWAT, 2007). A FIG. 8.11a ilustra o padrão. Esse voo foi o mais longo e durou 05m:47s. A FIG. 8.11b mostra com maiores detalhes uma região selecionada onde é possível observar o formato estimado semelhante a um osso. Isso acontece por causa das manobras agressivas. Assim, quando o VANT atinge o waypoint 4 da parte inferior direita, quase que imediatamente a aeronave tenta corrigir sua trajetória em busca do próximo waypoint (canto inferior esquerdo). Apesar disso, a missão foi cumprida satisfatoriamente e os erros permaneceram semelhantes aos obtidos com os demais experimentos.



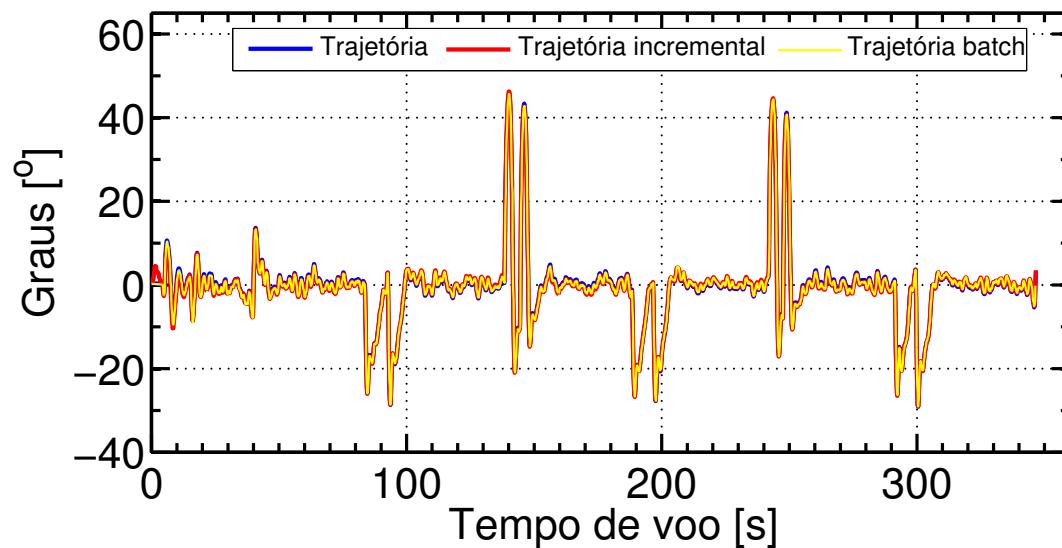
(a) Padrão lawn-mower (preto).



(b) Região aproximada. A trajetória batch (amarelo) é a mais suavizada.

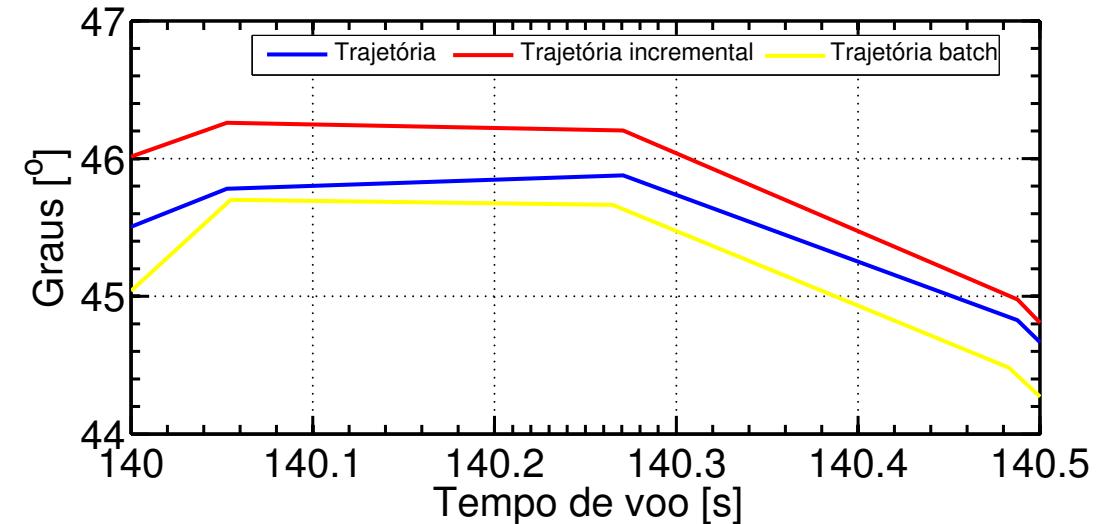
FIG. 8.11: Padrão lawn-mower (preto) e as trajetórias real (azul), incremental (vermelho) e batch (amarelo).

Rolamento (ϕ)



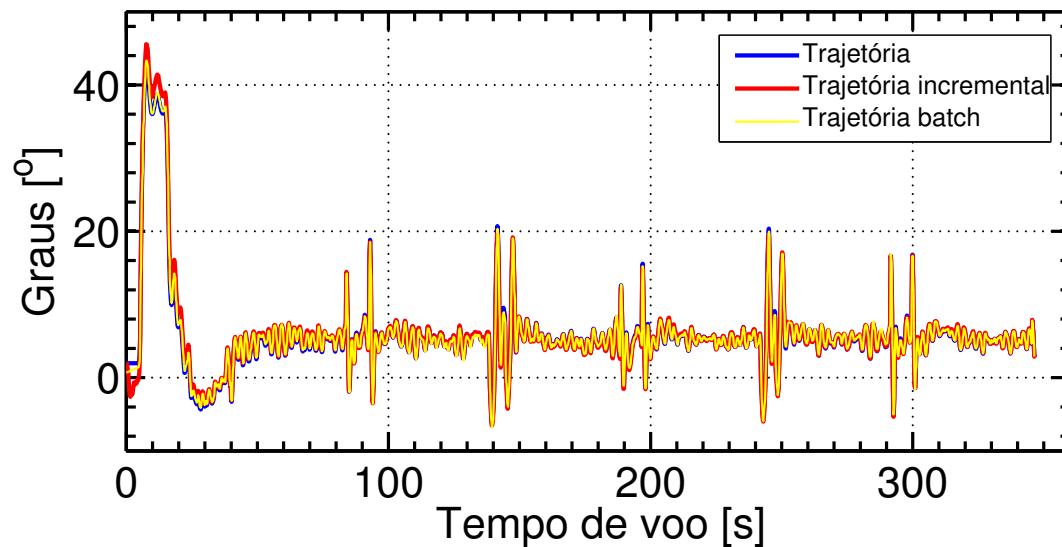
(a) Estimativa para o rolamento.

Rolamento (ϕ) – região aproximada



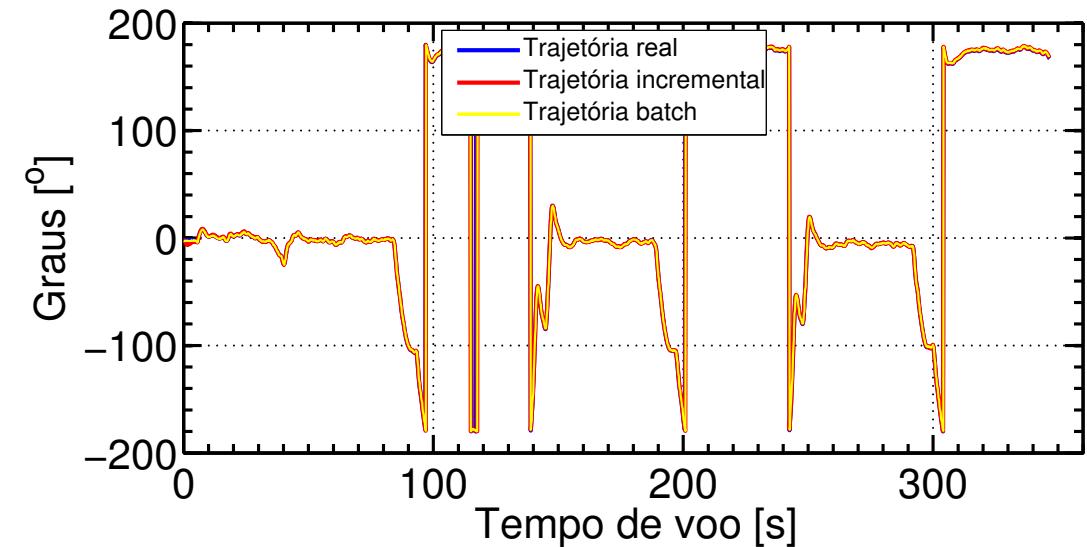
(b) Região aproximada para o rolamento.

Arfagem (θ)



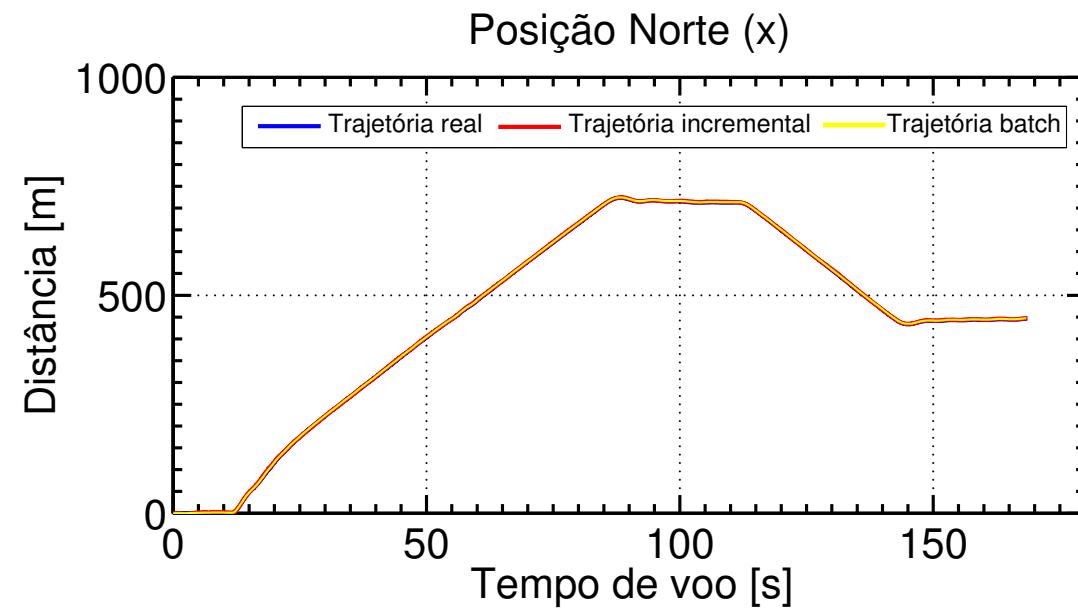
(c) Estimativa para a arfagem.

Guinada (ψ)

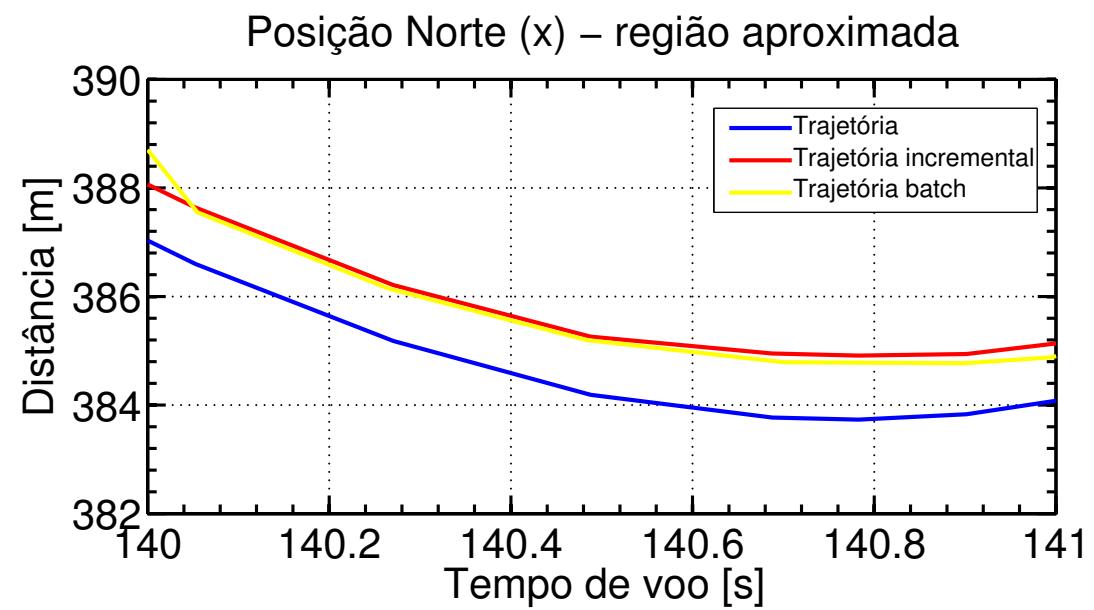


(d) Estimativa para a guinada.

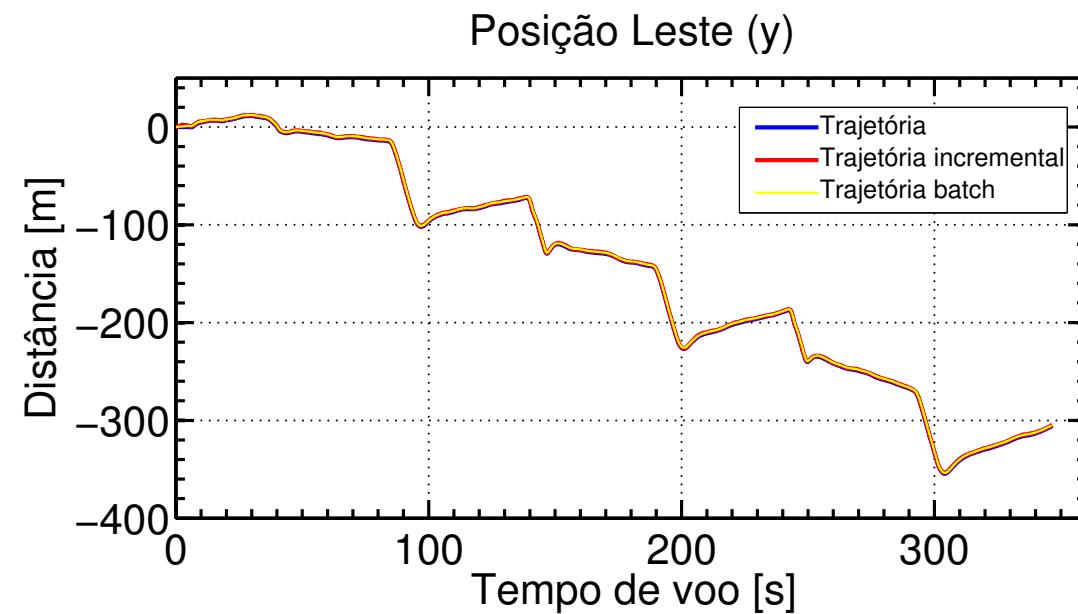
FIG. 8.12: Voo lawn-mower. Estimativas para a orientação.



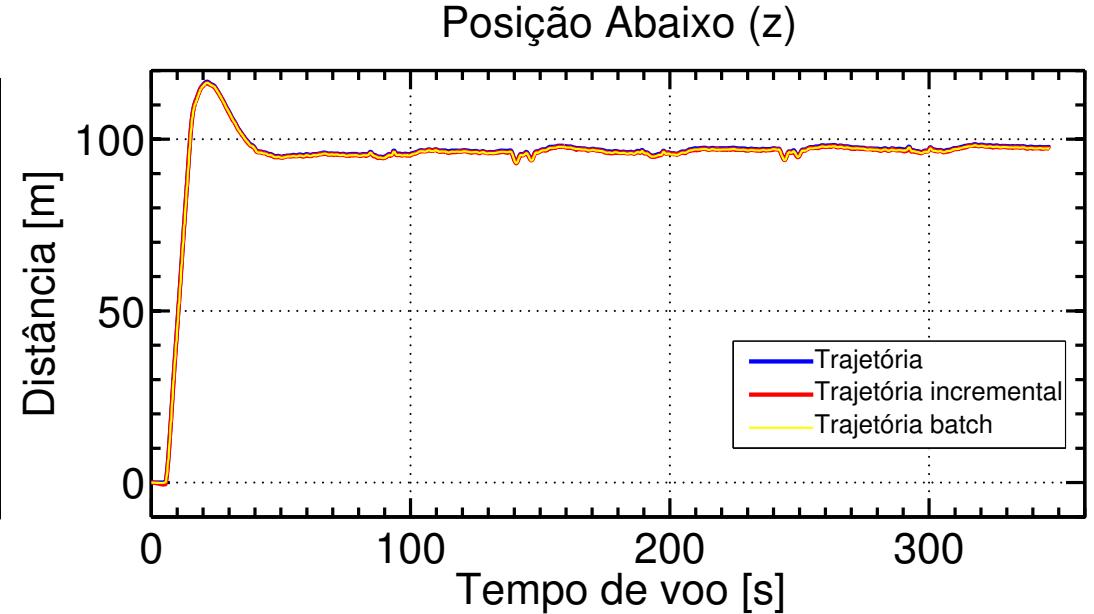
(a) Estimativa para a posição norte.



(b) Região aproximada para a posição norte.



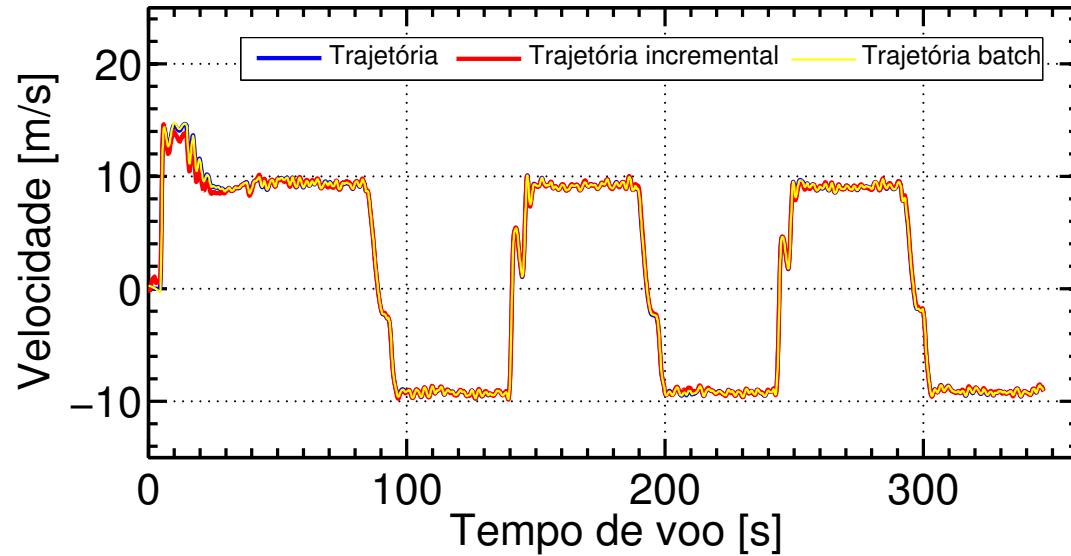
(c) Estimativa para a posição leste.



(d) Estimativa para a posição abaixo.

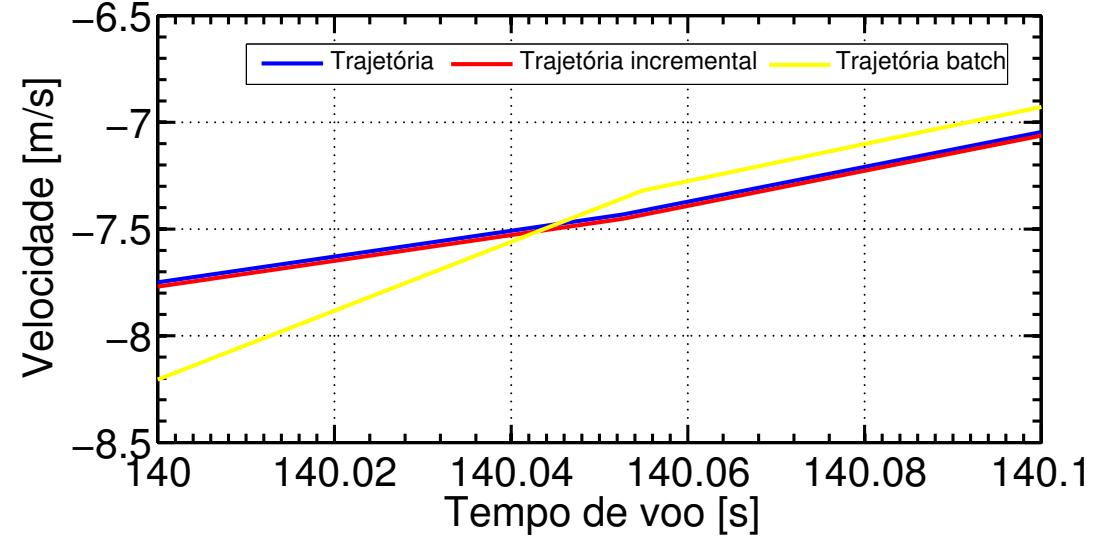
FIG. 8.13: Voo lawn-mower. Estimativas para a posição.

Velocidade Norte (x)



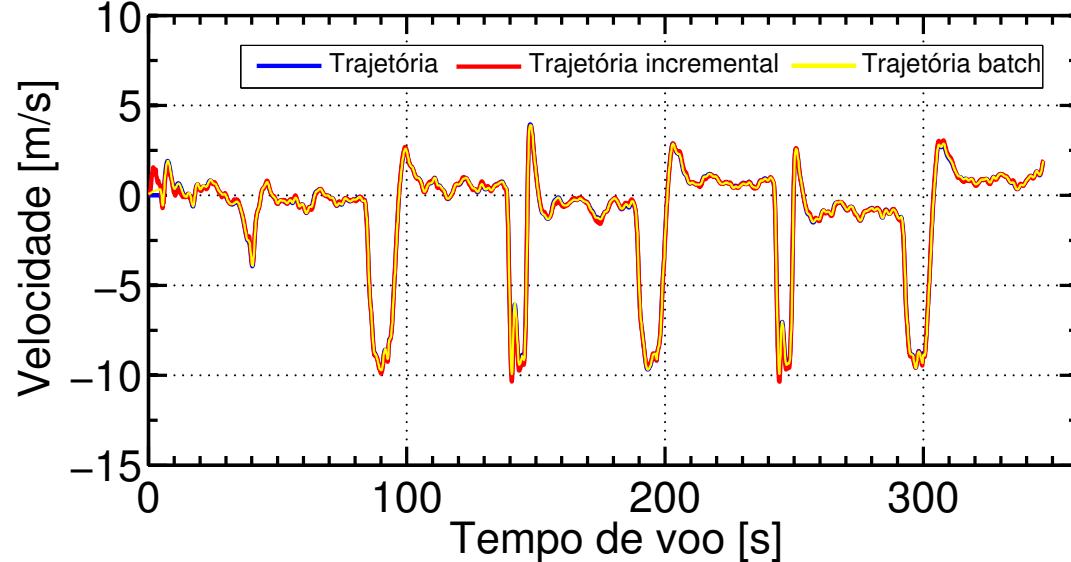
(a) Estimativa para a velocidade norte.

Velocidade Norte (x) – região aproximada



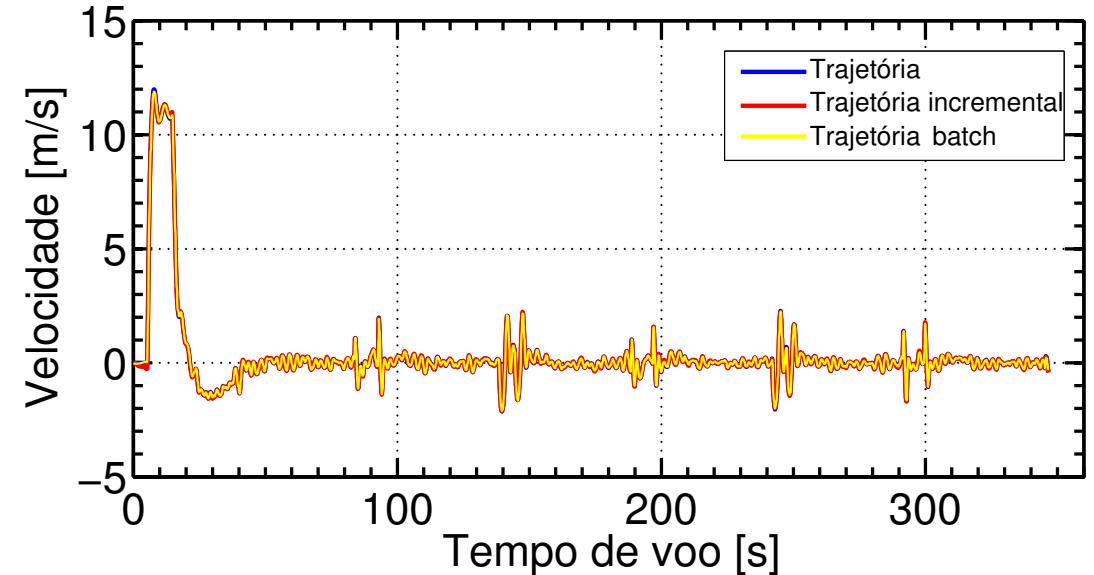
(b) Região aproximada para a velocidade norte.

Velocidade Leste (y)



(c) Estimativa para a velocidade leste.

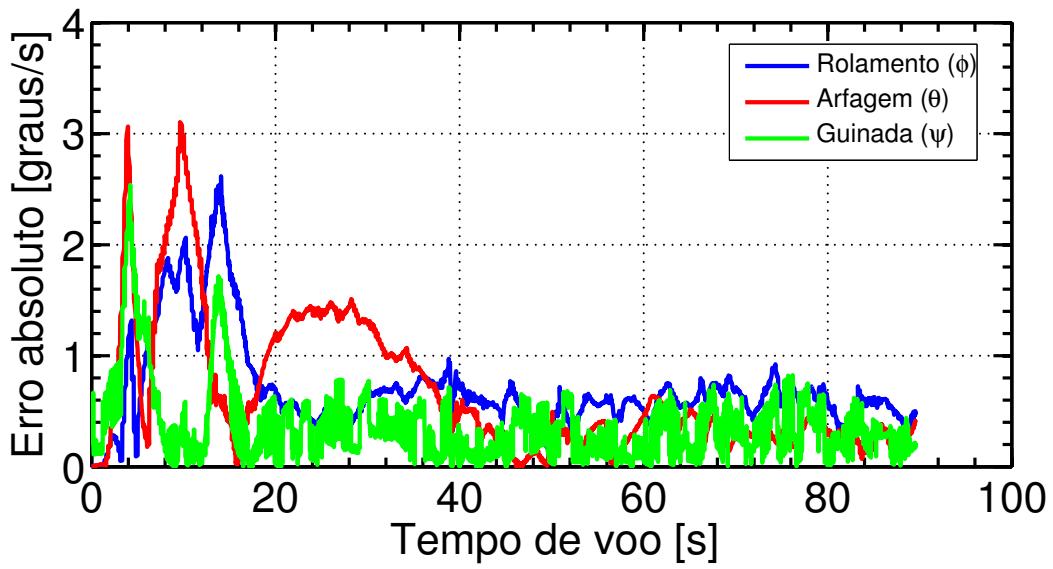
Velocidade Abaixo (z)



(d) Estimativa para a velocidade abaixo.

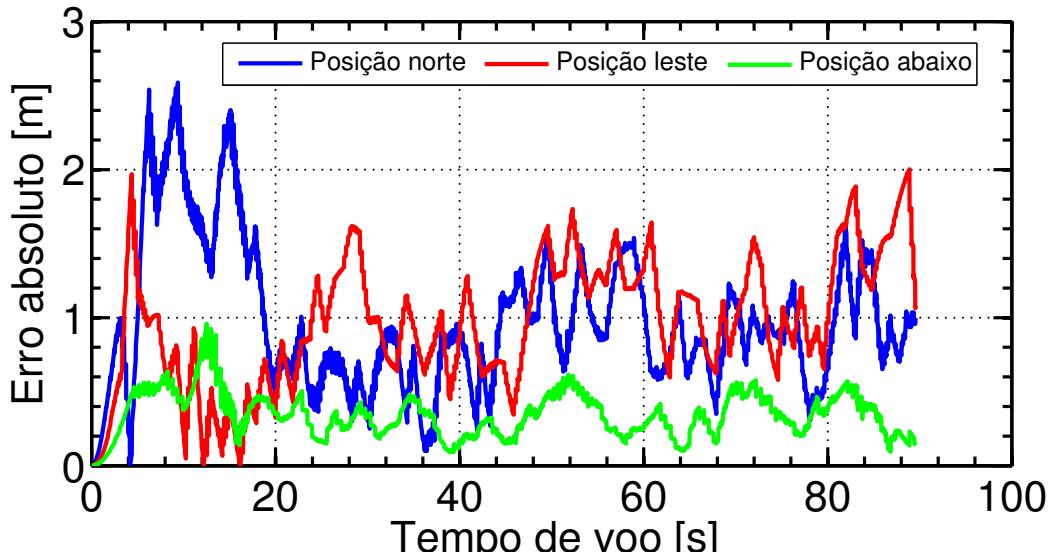
FIG. 8.14: Voo lawn-mower. Estimativas para a velocidade.

Erro absoluto da orientação



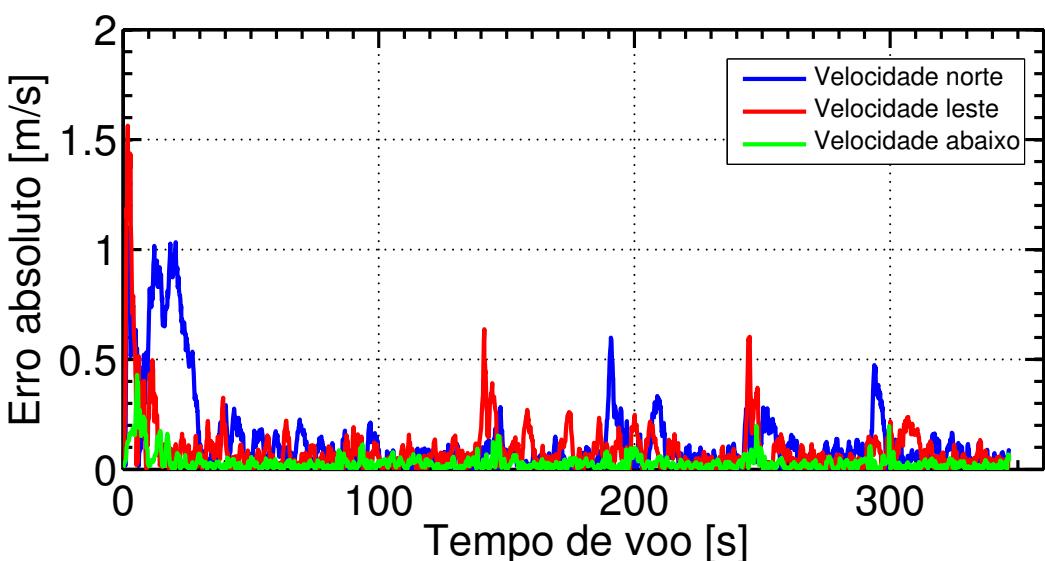
(a) Erro absoluto da orientação.

Erro absoluto das posições



(b) Erro absoluto da posição.

Erro absoluto das velocidades



(c) Erro absoluto da velocidade.

FIG. 8.15: Voo lawn-mower. Erros absolutos da orientação, posição e velocidade.

8.2 RESULTADOS CONSOLIDADOS DAS SIMULAÇÕES

Finalmente, a TAB. 8.2 apresenta os resultados consolidados. Apesar dos diferentes tipos de voo e de manobras executadas pelo VANT, o erro médio quadrático permaneceu semelhante entre os experimentos. Algumas estimativas incrementais apresentaram resultados muito próximos das sua correlata batch, evidenciando resultados satisfatórios e em tempo-real.

Experimento	Padrão	Tempo de voo [mm:ss.ff]	Algoritmo	Atitude [graus]			Posição [m]			Velocidade [m/s]		
				Rolam.	Arfag.	Guin.	N	E	D	V _N	V _E	V _D
1	Quadrado	02:48.57	Incremental	0,653	1,115	0,567	1,160	0,992	0,351	0,389	0,229	0,063
			Batch	0,467	0,518	0,333	1,081	0,944	0,312	0,076	0,053	0,036
2	Montanha-russa	01:29.55	Incremental	0,856	1,415	0,426	1,178	1,062	0,389	0,442	0,229	0,085
			Batch	0,613	0,614	0,196	1,128	1,043	0,330	0,088	0,051	0,055
3	Lawn-mower	05:46.88	Incremental	0,541	0,786	0,295	1,110	1,049	0,379	0,318	0,183	0,120
			Batch	0,415	0,293	0,259	1,067	0,989	0,330	0,076	0,052	0,040

TAB. 8.2: Resultados.

8.3 ENSAIO DE VOO

Após a realização das simulações, foi realizado um ensaio de voo no Clube de Aeromodelismo da Vila Militar (Deodoro - RJ). Nessa etapa, os objetivos eram:

- a) verificar a viabilidade do voo manual, sem a eletrônica embarcada. O propósito era alinhar os elevons, ajustar o centro de gravidade e testar o rádio controle resguardando os demais componentes.
- b) realizar um voo manual com a eletrônica embarcada. Desta vez, o propósito foi testar a telemetria em conjunto com a ECS e calibrar os sensores. (FIG. 8.16)
- c) realizar um voo autônomo. Finalmente, nesta etapa, verificariamos a viabilidade de um voo guiado pelo piloto automático, sem a interferência do operador.

As duas primeiras etapas transcorreram sem problemas. Para o voo autônomo, decidimos realizar um voo quadrado, por ser o mais simples, onde manobras mais arriscadas seriam evitadas. As duas primeiras tentativas de acionamento do piloto automático foram rejeitadas. Isso ocorria devido ao mecanismo interno de segurança que evitava o modo autônomo caso a aeronave não estivesse estabilizada e com um bom sinal de GPS. A aeronave estabilizada evita uma manobra brusca ao ser acionado o piloto automático. Sem o sinal de GPS, os erros das estimativas dos estados da aeronave crescem rapidamente (FIG. 8.17), inviabilizando qualquer possibilidade de voo autônomo.

Na terceira tentativa, conseguimos estabilizar a aeronave e realizar nosso primeiro voo autônomo. A aeronave cumpriu de maneira satisfatória a missão, percorrendo todos os waypoints do voo quadrado, sem observarmos instabilidades. Após essa primeira tentativa, realizamos mais um voo quadrado que apresentou resultados semelhantes.

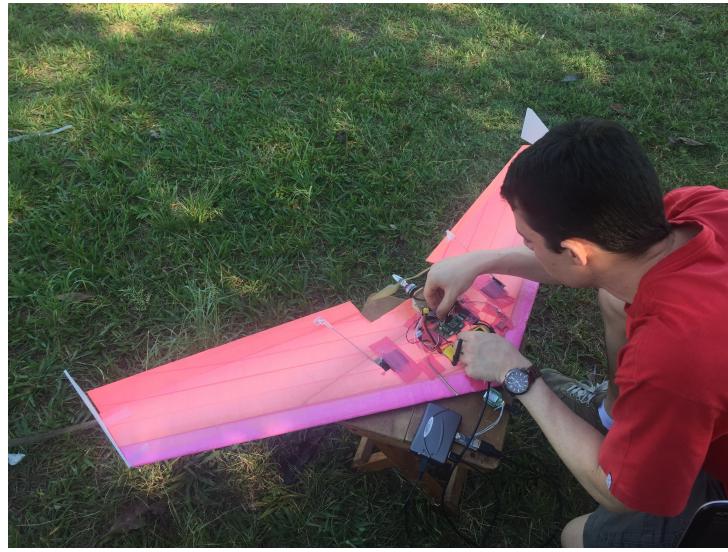


FIG. 8.16: Preparando a eletrônica embarcada para os voos manual e autônomo.

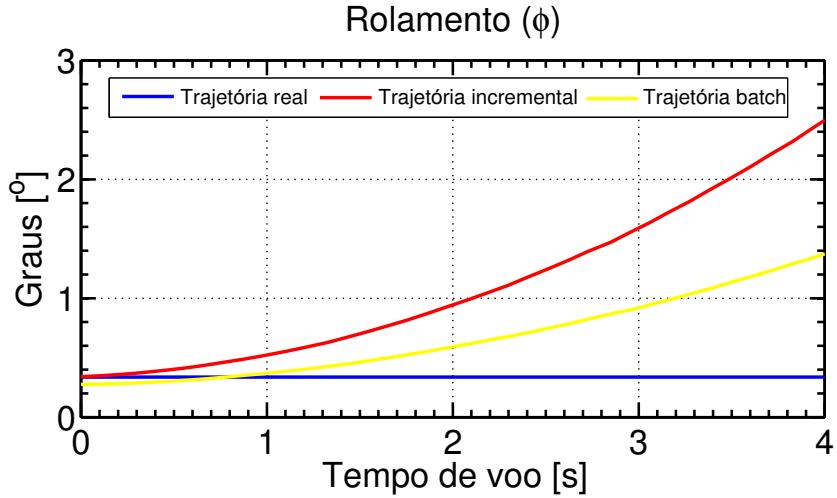


FIG. 8.17: Crescimento no erro do ângulo de rolamento quando não há auxílio do GPS.

8.3.1 ANÁLISE

Após o ensaio de voo, os dados armazenados no cartão de memória do piloto automático foram processados, gerando a trajetória em azul da FIG. 8.19. Em seguida, tentamos recriar o voo no ambiente simulado, obtendo a trajetória em vermelho da mesma figura. Como não dispunhamos de equipamentos de precisão que pudessem servir de referência, analisamos estatisticamente somente os dados da altitude, uma vez que esta foi definida constante em 100 m. Os resultados foram consolidados na TAB. 8.3. Observa-se que o voo real apresentou maiores oscilações e variações caracterizados pelo maior desvio padrão e pela diferença entre o máximo e o mínimo. Esse era o comportamento esperado, pois, o voo real apresenta diversas dificuldades e situações não comportados pelo simulador. Com relação às recriações das trajetórias, percebe-se que o voo simulado foi um pouco mais suave com curvas mais abertas, além de diferenças nas rotas. Diversos fatores po-



(a) VANT realizando o primeiro voo autônomo.



(b) Acompanhamento do voo quadrado pela ECS.

FIG. 8.18: Ensaio de voo com aconchamento da missão via estação de controle em solo.

dem explicar esses fatos: a presença do vento, a qualidade do modelo aerodinâmico, as simplificações nos modelos do movimento e dos sensores (consideramos apenas os ruídos brancos), a diferença nas condições iniciais (posição, orientação e velocidade). No entanto, consideramos que os objetivos foram atingidos, uma vez que, o voo autônomo foi conseguido no primeiro dia do ensaio de voo e sem acidentes.

Experimento	Altitude (m)			
	Média	Desvio padrão	Mínimo	Máximo
Voo real	94,5430	6,7825	81,9720	114,9900
Voo simulado	92,9395	5.1194	80,2630	104,6220

TAB. 8.3: Dados estatísticos da altitude (definida como 100 m) do voo real e simulado.



FIG. 8.19: Comparativo entre o voo real e o simulado.

9 CONSIDERAÇÕES FINAIS

Neste trabalho, apresentamos um mecanismo de fusão sensorial baseado numa técnica chamada suavização incremental via grafos-fatores. Mostramos que o problema da navegação de um veículo aéreo não tripulado pode ser visto como um grafo contendo dois tipos de nós: as variáveis que representam os estados da aeronave a serem estimados (posição, orientação e velocidade) e os fatores que representam os sensores. As arestas ligam esses nós e caracterizam o erro entre o modelo e a medição realizada pelo sensor. O objetivo é minimizar esses erros, de maneira a obter a melhor estimativa para os estados. Nesse caso, o problema pode ser transformado numa otimização via método dos mínimos quadrados. Uma vez que o grafo cresce rapidamente ao longo do tempo, onde a otimização de todo o grafo, passa a ser inviável. O problema foi resolvido com o uso de atualizações incrementais, onde somente parte do grafo é recalculado, ganhando performance. Para auxiliar ainda mais na velocidade, utilizamos o conceito da pré-integração que acumula os dados da unidade de medida inercial, diminuindo a quantidade de nós adicionados ao grafo.

Além disso, elaboramos um VANT de asa fixa no formato delta. Essa asa é largamente utilizada, consagrada e possui modelos aerodinâmicos disponíveis em alguns simuladores de voo. Na elaboração, utilizamos componentes eletrônicos bastante leves e pequenos, podendo ser adaptados para plataformas de menor porte como os microVANTs. Para auxiliar no processo da navegação, foi escolhido um conjunto de sensores composto por um acelerômetro, um girômetro, um magnetômetro, um barômetro, um sensor de pressão diferencial e um GPS. Cada um desses sensores contribui de alguma forma na obtenção das estimativas dos estados da aeronave.

Para a validação do método, propomos o uso de um simulador de voo, onde implementamos um plugin para realizar a fusão dos dados dos sensores. O plugin recebe os dados dos sensores provenientes do simulador de voo, adiciona ruído branco, gera o grafo-fator e finalmente, executa a suavização incremental. O protocolo de comunicações MAVLink envia as estimativas para uma estação de controle em solo e para o piloto automático, que retorna as saídas do controle da atitude e posição. Nesse processo ininterrupto é possível realizar um voo simulado. Dessa forma, criamos três missões para a aeronave: um voo quadrado, um voo montanha-russa e um voo lawn-mower, útil em tarefas de busca

e varredura de áreas. Apesar dos diferentes tipos de voo e de manobras executadas pelo VANT, o erro médio quadrático permaneceu semelhante entre os experimentos. Verificou-se que algumas estimativas incrementais apresentaram resultados muito próximos das estimativas batch, evidenciando resultados satisfatórios e em tempo-real.

Finalmente, foi realizado um ensaio de voo no Clube de Aeromodelismo da Vila Militar, onde a aeronave foi ajustada e os sensores calibrados antes de tentarmos o voo autônomo. Após esses ajustes, conseguimos realizar nosso primeiro voo controlado por piloto automático de maneira satisfatória e sem problemas de avarias com a aeronave ou os equipamentos. Após o processamento dos dados dos sensores, recriamos o voo no simulador e percebemos algumas diferenças entre ambos. Dentre os fatores estão a presença de vento, a qualidade do modelo aerodinâmico, as simplificações nos modelos do movimento e dos sensores, e a dificuldade de recriar as mesmas condições iniciais para os voos. Outro fator a ser comentado, é a dependência do GPS como forma de evitar o rápido crescimento dos erros das estimativas. Os outros sensores não conseguem realizar esse papel, pois o barômetro só observa a altitude e o magnetômetro, o ângulo de guinada. No entanto, dadas as dificuldades e os desafios, conseguimos um voo autônomo no primeiro dia de ensaio de voo, caracterizando a viabilidade do método, das simulações e da elaboração do VANT.

9.1 TRABALHOS FUTUROS

As propostas apresentadas neste trabalho, apesar de práticas e funcionais, podem ser melhoradas, expandindo as suas funcionalidades e qualidades. Podemos destacar:

- Agregar novos sensores que possam evitar a dependência do GPS. Diversas pesquisas nesse sentido vêm sendo realizadas, com destaque para as soluções envolvendo visão computacional. Esses sensores são baratos e fáceis de serem agregados em um VANT. A desvantagem é que muitas das técnicas de visão computacional requerem um maior poder computacional. Outro sensor que pode ser considerado é o de fluxo ótico. Com este sensor é possível estimar os vetores de velocidade, mediante a análise dos pixels entre duas imagens consecutivas, auxiliando as estimativas na ausência do GPS.
- Implementar melhorias nos modelos dos sensores considerando outros tipos de características inerentes a cada um deles. Uma possibilidade é o uso da variância de

Allan para inferir outras informações além dos ruídos brancos considerados.

- Implementar melhorias no modelo aerodinâmico da aeronave de maneira a ficar mais próximo da realidade.
- Considerar outros fatores externos nas simulações, principalmente a influência dos ventos.

10 REFERÊNCIAS BIBLIOGRÁFICAS

- AGARWAL, P. e OLSON, E. Variable reordering strategies for slam. Em *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, págs. 3844–3850. IEEE, 2012.
- ALVES, O. D. L. F. Planejamento e controle de missões de um vane de asa fixa. Dissertação de Mestrado, 2010.
- ANGONESE, A. T. Estação de controle em solo com funcionalidades de voo múltiplo para vants. Dissertação de Mestrado, 2013a.
- ANGONESE, A. e FERREIRA ROSA, P. Ground control station for multiple uavs flight simulation. Em *Robotics Symposium and Competition (LARS/LARC), 2013 Latin American*, págs. 136–141, Oct 2013b.
- APM. Ardupilot mega - multiplataform autopilot, 2014. URL <http://ardupilot.com/>.
- BISHOP, C. M. e OTHERS. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.
- BRONZ, M., CONDOMINES, J.-P., HATTENBERGER, G. e OTHERS. Development of an 18cm micro air vehicle: Quark. Em *Proceedings of the International Micro Air Vehicle Conference and Flight Competition IMAV 2013*, 2013a.
- BRONZ, M., HATTENBERGER, G. e MOSCHETTA, J.-M. Development of a long endurance mini-uav: Eternity. *International Journal of Micro Air Vehicles*, 5(4):261–272, 2013b.
- BRYSON, M., JOHNSON-ROBERSON, M. e SUKKARIEH, S. Airborne smoothing and mapping using vision and inertial sensors. Em *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, págs. 2037–2042. IEEE, 2009.
- CAI, G., LUM, K.-Y., CHEN, B. M. e LEE, T. H. A brief overview on miniature fixed-wing unmanned aerial vehicles. Em *Control and Automation (ICCA), 2010 8th IEEE International Conference on*, págs. 285–290. IEEE, 2010.
- CALTABIANO, D., MUSCATO, G., ORLANDO, A., FEDERICO, C., GIUDICE, G. e GUERRIERI, S. Architecture of a uav for volcanic gas sampling. Em *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on*, volume 1, págs. 6 pp.–744, Sept 2005.
- CARVALHO, D. D. A. C. Desenvolvimento de uma plataforma hardware in the loop para simulação de uma estação de controle em solo de um vane de pequeno porte utilizando x-plane. Em *Congresso Brasileiro de Automática, CBA 2012*. CBA, 2012.

- COOK, M. V. *Flight dynamics principles: a linear systems approach to aircraft stability and control*. Butterworth-Heinemann, 2012.
- COWELL, R. G. *Probabilistic networks and expert systems: Exact computational methods for Bayesian networks*. Springer Science & Business, 2006.
- CRASSIDIS, J. L., MARKLEY, F. L. e CHENG, Y. Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007.
- DA PAIXÃO, R., MAGALHAES NETO, J. e FERREIRA ROSA, P. Prototyping a basic mems attitude heading and reference system. Em *Industrial Technology (ICIT), 2011 IEEE International Conference on*, págs. 232–236, 2011.
- DARPA. All Source Positioning and Navigation (ASPN). Broad Agency Annoucement, Novembro 2010.
- DAVIS, T. A., GILBERT, J. R., LARIMORE, S. I. e NG, E. G. Algorithm 836: Colamd, a column approximate minimum degree ordering algorithm. *ACM Transactions on Mathematical Software (TOMS)*, 30(3):377–380, 2004.
- DE MAESSCHALCK, R., JOUAN-RIMBAUD, D. e MASSART, D. L. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.
- DELLAERT, F. e KAESZ, M. Square root sam: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203, 2006.
- DOS SANTOS, S. R. B., JÚNIOR, C. L. N., JUNIOR, S. N. G., BITTAR, A. e DE OLIVEIRA, N. M. F. Experimental framework for evaluation of guidance and control algorithms for uavs. 2011.
- EUSTICE, R. M., SINGH, H. e LEONARD, J. J. Exactly sparse delayed-state filters for view-based slam. *Robotics, IEEE Transactions on*, 22(6):1100–1114, 2006.
- FARRELL, J. *Aided navigation: GPS with high rate sensors*. McGraw-Hill New York, 2008.
- FINEP. Chamada Pública/ MCT/FINEP - CT-AERO - VANT 01/2009, Dezembro 2009.
- FOLKESSON, J., JENSFELT, P. e CHRISTENSEN, H. I. Graphical slam using vision and the measurement subspace. Em *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, págs. 325–330. IEEE, 2005.
- FOSSEN, T. I. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.
- FOWLER, K. R. e SCHMALZEL, J. L. Sensors: The first stage in the measurement chain. *Instrumentation & Measurement Magazine, IEEE*, 7(3):60–65, 2004.

- GAO, W., ZHANG, Y. e SUN, Q. An asynchronous fusion algorithm of the sins/gps/cns based on factor graph. Em *Control Conference (CCC), 2013 32nd Chinese*, págs. 4995–4999. IEEE, 2013.
- GOLUB, G. H. e VAN LOAN, C. F. *Matrix computations*, volume 3. JHU Press, 2012.
- GOPPERT, J. M. An adaptable, low cost test-bed for unmanned vehicle systems research, 2011.
- GRASSIA, F. S. Practical parameterization of rotations using the exponential map. *Journal of graphics tools*, 3(3):29–48, 1998.
- GREWAL, M. S., WEILL, L. R. e ANDREWS, A. P. *Global positioning systems, inertial navigation, and integration*. John Wiley & Sons, 2007.
- GRISSETTI, G., KUMMERLE, R., STACHNISS, C., FRESE, U. e HERTZBERG, C. Hierarchical optimization on manifolds for online 2d and 3d mapping. Em *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, págs. 273–278. IEEE, 2010.
- GROVES, P. D. *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech House, 2013.
- GUERRERO, F., MARTINEZ, V., GARCIA, O. e MARTINEZ, D. Mini airplane: Design, aerodynamic modeling and stability. Em *Electrical Engineering, Computing Science and Automatic Control (CCE), 2014 11th International Conference on*, págs. 1–6. IEEE, 2014.
- HATTENBERGER, G., CAYEZ, G., ROBERTS, G. e OTHERS. Flight tests for meteorological studies with mav. Em *Proceedings of the International Micro Air Vehicle Conference and Flight Competition IMAV 2013*, 2013.
- HUANG, G. P., MOURIKIS, A. I. e ROUMELIOTIS, S. I. An observability-constrained sliding window filter for slam. Em *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, págs. 65–72. IEEE, 2011.
- INDELMAN, V., WILLIAMS, S., KAESS, M. e DELLAERT, F. Factor graph based incremental smoothing in inertial navigation systems. Em *Information Fusion (FUSION), 2012 15th International Conference on*, págs. 2154–2161. IEEE, 2012.
- INDELMAN, V., WILLIAMS, S., KAESS, M. e DELLAERT, F. Information fusion in navigation systems via factor graph based incremental smoothing. *Robotics and Autonomous Systems*, 61(8):721–738, 2013.
- JOVANOVIC, M. e STARCEVIC, D. Software architecture for ground control station for unmanned aerial vehicle. Em *Computer Modeling and Simulation, 2008. UKSIM 2008. Tenth International Conference on*, págs. 284–288, April 2008.

- KAESS, M., ILA, V., ROBERTS, R. e DELLAERT, F. The bayes tree: An algorithmic foundation for probabilistic robot mapping. *Algorithmic Foundations of Robotics IX*, págs. 157–173, 2011a.
- KAESS, M., JOHANNSSON, H., ROBERTS, R., ILA, V., LEONARD, J. e DELLAERT, F. isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. Em *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, págs. 3281–3288. IEEE, 2011b.
- KAESS, M., JOHANNSSON, H., ROBERTS, R., ILA, V., LEONARD, J. J. e DELLAERT, F. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.
- KAESS, M., RANGANATHAN, A. e DELLAERT, F. isam: Incremental smoothing and mapping. *Robotics, IEEE Transactions on*, 24(6):1365–1378, 2008.
- KALMAN, R. E. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.
- KAPLAN, E. D. e HEGARTY, C. J. *Understanding GPS: principles and applications*. Artech house, 2005.
- KINGSTON, D. B. e BEARD, R. W. Real-time attitude and position estimation for small uav's using low-cost sensors. Em *AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*, págs. 2004–6488. sn, 2004.
- KOLLER, D. e FRIEDMAN, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- KSCHISCHANG, F. R., FREY, B. J. e LOELIGER, H.-A. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519, 2001.
- KUMMERLE, R., GRISETTI, G., STRASDAT, H., KONOLIGE, K. e BURGARD, W. g2o: A general framework for graph optimization. Em *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, págs. 3607–3613. IEEE, 2011.
- LANGE, S., SUNDERHAUF, N. e PROTZEL, P. Incremental smoothing vs. filtering for sensor fusion on an indoor uav. Em *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, págs. 1773–1778. IEEE, 2013.
- LATIF, Y. e NEIRA, J. Go straight, turn right: Pose graph reduction through trajectory segmentation using line segments. Em *Mobile Robots (ECMR), 2013 European Conference on*, págs. 144–149. IEEE, 2013.
- LIZARRAGA, M., ELKAIM, G. H. e CURRY, R. Slugs uav: A flexible and versatile hardware/software platform for guidance navigation and control research. Em *American Control Conference (ACC), 2013*, págs. 674–679. IEEE, 2013.
- LU, F. e MILIOS, E. Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4):333–349, 1997.

- LUPTON, T. e SUKKARIEH, S. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *Robotics, IEEE Transactions on*, 28(1):61–76, 2012.
- MADSEN, K., NIELSEN, H. B. e TINGLEFF, O. Methods for non-linear least squares problems. 2004.
- MAGNUSSON, T. State estimation of uav using extended kalman filter. Dissertaçāo de Mestrado, Linköping UniversityLinköping University, Automatic Control, The Institute of Technology, 2013.
- MAO, G., DRAKE, S. e ANDERSON, B. D. Design of an extended kalman filter for uav localization. Em *Information, Decision and Control, 2007. IDC'07*, págs. 224–229. IEEE, 2007.
- MAUS, S., MACMILLAN, S., MCLEAN, S., HAMILTON, B., THOMSON, A., NAIR, M. e ROLLINS, C. The us/uk world magnetic model for 2010-2015. 2010.
- MCGHEE, R., BACHMANN, E. R. e ZYDA, M. J. Rigid body dynamics, inertial reference frames, and graphics coordinate systems: A. *IEEE Computer Graphics and Applications*, 8(1):19–27, 1988.
- MEIER, L. Mavlink: Micro air vehicle communication protocol, 2009.
- MEIER, L., TANSKANEN, P., FRAUNDORFER, F. e POLLEFEYS, M. Pixhawk: A system for autonomous flight using onboard computer vision. Em *Robotics and automation (ICRA), 2011 IEEE international conference on*, págs. 2992–2997. IEEE, 2011.
- MEYER, A. X-plane: Ultra-realistic flight simulation, 2014. URL <http://www.x-plane.com/>.
- MINISTÉRIO DA DEFESA. Estratégia Nacional de Defesa, Dezembro 2008.
- MITCHELL, H. B. *Data fusion: concepts and ideas*. Springer, 2012.
- MOURIKIS, A. I. e ROUMELIOTIS, S. I. A multi-state constraint kalman filter for vision-aided inertial navigation. Em *Robotics and Automation, 2007 IEEE International Conference on*, págs. 3565–3572. IEEE, 2007.
- MOURIKIS, A. I. e ROUMELIOTIS, S. I. A dual-layer estimator architecture for long-term localization. Em *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, págs. 1–8. IEEE, 2008.
- MUELLER, T. J. e DELAURIER, J. D. Aerodynamics of small vehicles. *Annual Review of Fluid Mechanics*, 35(1):89–111, 2003.
- NAUDIN, J.-L. Experimental researches and development of autopilots for aerial platforms, 2014. URL <https://github.com/jlnaudin>.
- NAVSTIK. Micro navigation and control board, 2014. URL <http://navstik.org/>.

NETO, J., RODRIGUES, L., MOREIRA, E., DOS SANTOS, J. e ROSA, P. Uma missao de monitoramento para o projeto vant-ime: Operacao regiao serranarj, 2011.

NOCEDAL, J. e WRIGHT, S. J. Numerical optimization 2nd. 2006.

NORDSJO, A. Target tracking based on kalman-type filters combined with recursive estimation of model disturbances. Em *Radar Conference, 2005 IEEE International*, págs. 115–120. IEEE, 2005.

OUSINGSAWAT, J. e EARL, M. G. Modified lawn-mower search pattern for areas comprised of weighted regions. Em *American Control Conference, 2007. ACC'07*, págs. 918–923. IEEE, 2007.

PANDAPILOT. Navstik aerial middleware and flight control stack, 2014. URL https://github.com/navstik/pandapilot_v4.

PASTOR, E., LOPEZ, J. e ROYO, P. Uav payload and mission control hardware/software architecture. *Aerospace and Electronic Systems Magazine, IEEE*, 22(6):3–8, 2007.

PEARL, J. Bayesian networks. Em *The handbook of brain theory and neural networks*, págs. 149–153. MIT Press, 1998.

PENN, T. R. All source sensor integration using an extended kalman filter. 2012.

PÖLLÄNEN, R., TOIVONEN, H., PERÄJÄRVI, K., KARHUNEN, T., ILANDER, T., LEHTINEN, J., RINTALA, K., KATAJAINEN, T., NIEMELÄ, J. e JUUSELA, M. Radiation surveillance using an unmanned aerial vehicle. *Applied Radiation and Isotopes*, 67(2):340 – 344, 2009. ISSN 0969-8043. URL <http://www.sciencedirect.com/science/article/pii/S0969804308004831>.

PLUMBLEY, M. D. Lie group methods for optimization with orthogonality constraints. Em *Independent Component Analysis and Blind Signal Separation*, págs. 1245–1252. Springer, 2004.

REUDER, J., BRISSET, P., JONASSEN, M., MÜLLER, M. e MAYER, S. Sumo: A small unmanned meteorological observer for atmospheric boundary layer research. Em *IOP Conference Series: Earth and Environmental Science*, volume 1, pág. 012014. IOP Publishing, 2008.

RIBEIRO, L. R. e OLIVEIRA, N. M. F. Uav autopilot controllers test platform using matlab/simulink and x-plane. Em *Frontiers in Education Conference (FIE), 2010 IEEE*, págs. S2H–1. IEEE, 2010.

SHILOV, K. The next generation design of autonomous mav flight control system smartap. Em *IMAV 2014: International Micro Air Vehicle Conference and Competition 2014, Delft, The Netherlands, August 12-15, 2014*. Delft University of Technology, 2014.

SIBLEY, G., MATTHIES, L. e SUKHATME, G. Sliding window filter with application to planetary landing. *Journal of Field Robotics*, 27(5):587–608, 2010.

- SKOGLUND, M. A., GUSTAFSSON, F. e JONSSON, K. Modeling and sensor fusion of a remotely operated underwater vehicle. Em *Information Fusion (FUSION), 2012 15th International Conference on*, págs. 947–954. IEEE, 2012.
- SMITH, D. e SINGH, S. Approaches to multisensor data fusion in target tracking: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 18(12):1696–1710, 2006.
- SUGER, B., DIEGO TIPALDI, G., SPINELLO, L. e BURGARD, W. An approach to solving large-scale slam problems with a small memory footprint. Em *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, págs. 3632–3637, May 2014.
- SUNDERHAUF, N., OBST, M., WANIELIK, G. e PROTZEL, P. Multipath mitigation in gnss-based localization using robust optimization. Em *Intelligent Vehicles Symposium (IV), 2012 IEEE*, págs. 784–789. IEEE, 2012a.
- SUNDERHAUF, N. e PROTZEL, P. Towards robust graphical models for gnss-based localization in urban environments. Em *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*, págs. 1–6. IEEE, 2012b.
- SUNDERHAUF, N., OBST, M., LANGE, S., WANIELIK, G. e PROTZEL, P. Switchable constraints and incremental smoothing for online mitigation of non-line-of-sight and multipath effects. Em *Intelligent Vehicles Symposium (IV), 2013 IEEE*, págs. 262–268. IEEE, 2013.
- SUZUKI, T., MIYOSHI, D., MEGURO, J., AMANO, Y., HASHIZUME, T., SATO, K. e TAKIGUCHI, J. Real-time hazard map generation using small unmanned aerial vehicle. Em *SICE Annual Conference, 2008*, págs. 443–446, Aug 2008.
- SYSTEMS, E. T. Sensor de pressão dinâmica, 2014. URL <http://www.eagletreesystems.com/>.
- TARDIF, J.-P., GEORGE, M., LAVERNE, M., KELLY, A. e STENTZ, A. A new approach to vision-aided inertial navigation. Em *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, págs. 4161–4168. IEEE, 2010.
- THRUN, S., BURGARD, W. e FOX, D. *Probabilistic Robotics*. Intelligent robotics and autonomous agents series. Mit Press, 2005. ISBN 9780262201629. URL http://books.google.com.br/books?id=k_y0QgAACAAJ.
- VALIENTE, G. *Algorithms on trees and graphs*. Springer, 2002.
- WANG, B., CHEN, X., WANG, Q., LIU, L., ZHANG, H. e LI, B. Power line inspection with a flying robot. Em *Applied Robotics for the Power Industry (CARPI), 2010 1st International Conference on*, págs. 1–6, Oct 2010.
- WINKVIST, S., RUSHFORTH, E. e YOUNG, K. Towards an autonomous indoor aerial inspection vehicle. *Industrial Robot: An International Journal*, 40(3):196–207, 2013. URL <http://dx.doi.org/10.1108/01439911311309870>.

YANNAKAKIS, M. Computing the minimum fill-in is np-complete. *SIAM Journal on Algebraic Discrete Methods*, 2(1):77–79, 1981.

YONGLIANG, W., TIANMIAO, W., JIANHONG, L., CHAOLEI, W. e CHEN, Z. Attitude estimation for small helicopter using extended kalman filter. Em *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*, págs. 577–581. IEEE, 2008.

ZURICH, E. Qgroundcontrol: Ground control station for small air land water autonomous unmanned systems, 2013.

11 APÊNDICES

11.1 APÊNDICE A: OTIMIZAÇÃO VIA MÍNIMOS QUADRADOS

Neste apêndice elaboramos mais detalhadamente os problemas de otimização descritos no Capítulo 3. Vimos que na otimização via mínimos quadrados, a função objetivo é representada por uma soma de termos quadráticos, resultando no seguinte problema:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} F(\mathbf{x}) = \arg \min_{\mathbf{x}} \frac{1}{2} \mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) \quad (11.1)$$

Como a função objetivo pode ser linear ou não linear, e podemos ponderar ou não as informações, obtemos quatro tipos de problemas: (a) os lineares, (b) os lineares ponderados, (c) os não-lineares e (d) os não-lineares ponderados.

11.1.1 PROBLEMAS LINEARES

Nos casos mais simples, todas as funções f_i irão depender linearmente de \mathbf{x} assumindo assim a seguinte forma:

$$f_i(\mathbf{x}) = a_{i,1}x_1 + a_{i,2}x_2 \dots a_{i,n}x_n - b_i \quad (11.2)$$

Se agruparmos todos os coeficientes $a_{i,n}$ na matriz \mathbf{A} e todas as constantes b_i no vetor \mathbf{b} , vetorialmente teremos

$$\mathbf{f}(\mathbf{x}) = \mathbf{Ax} - \mathbf{b} \quad (11.3)$$

Lembrando que o mínimo \mathbf{x}^* é obtido quando a derivada primeira é igual a zero:

$$\begin{aligned}
F'(\mathbf{x}^*) &= 0 \\
\frac{\partial F}{\partial \mathbf{x}} &= 0 \\
\frac{\partial(\frac{1}{2}\mathbf{f}^\top \mathbf{f})}{\partial \mathbf{x}} &= 0 \\
\mathbf{f}^\top \frac{\partial \mathbf{f}}{\partial \mathbf{x}} &= 0 \\
(\mathbf{A}\mathbf{x}^* - \mathbf{b})^\top \frac{\partial(\mathbf{A}\mathbf{x}^* - \mathbf{b})}{\partial \mathbf{x}} &= 0 \\
(\mathbf{A}\mathbf{x}^* - \mathbf{b})^\top \mathbf{A} &= 0 \\
\mathbf{A}^\top (\mathbf{A}\mathbf{x}^* - \mathbf{b}) &= 0 \\
\mathbf{A}^\top \mathbf{A}\mathbf{x}^* - \mathbf{A}^\top \mathbf{b} &= 0 \\
\mathbf{A}^\top \mathbf{A}\mathbf{x}^* &= \mathbf{A}^\top \mathbf{b}
\end{aligned} \tag{11.4}$$

Para obtermos o valor de \mathbf{x}^* , devemos resolver a EQ. 11.4, conhecida como equação normal. Obviamente, a maneira mais simples é o cálculo direto de $\mathbf{x}^* = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$. Infelizmente, nem sempre $\mathbf{A}^\top \mathbf{A}$ pode ser invertida. Além disso, a inversão de matrizes é uma operação lenta. As melhores alternativas encontram-se no uso das decomposições matriciais como a decomposição QR e a decomposição *Cholesky*. Outras decomposições como a decomposição em valores singulares (SVD) também podem ser utilizadas. No entanto, esta não apresenta papel importante no desenvolver do trabalho.

Na decomposição QR, a matriz \mathbf{A} é decomposta em uma matriz ortogonal \mathbf{Q} e numa matriz triangular superior \mathbf{R} , ou seja, $\mathbf{A} = \mathbf{QR}$. A solução \mathbf{x}^* é dada por $\mathbf{x}^* = \mathbf{R}^{-1} \mathbf{Q}^\top \mathbf{b}$. Este método evita o cálculo de $\mathbf{A}^\top \mathbf{A}$ e a sua inversão, o que pode tornar-se instável numericamente. Já na decomposição *Cholesky*, fatora-se a matriz $\mathbf{A}^\top \mathbf{A}$ em \mathbf{LL}^\top , onde \mathbf{L} é uma matriz triangular inferior. Em seguida resolve-se $\mathbf{Ly} = \mathbf{A}^\top \mathbf{B}$ para \mathbf{y} e, finalmente, calcula-se \mathbf{x}^* resolvendo $\mathbf{L}^\top \mathbf{x}^* = \mathbf{y}$. Em comparação com a decomposição QR, *Cholesky* é mais rápida, porém numericamente mais instável pois $\mathbf{A}^\top \mathbf{A}$ precisa ser calculada.

Exemplo 11.1. Considere uma aeronave realizando um voo no plano \mathbb{R}^1 com aceleração constante. Sabemos que a posição y da aeronave em função do tempo t pode ser descrita por uma função quadrática

$$y(t) = \frac{1}{2}at^2 + v_0t + y_0 \tag{11.5}$$

onde a é a aceleração, v_0 é a velocidade inicial, e y_0 é a posição inicial. Infelizmente não sabemos esses três parâmetros. Contudo, com o uso de um simples radar móvel operando à $1Hz$, fizemos 10 medições e anotamos os resultados na TAB. 11.1. Obviamente o radar possui limitações, ou seja, as medições y_r possuem ruído branco gaussiano $w \sim N(0, \sigma^2 = 9m)$.

$$y_r(t) = y(t) + w \quad (11.6)$$

t [s]	1	2	3	4	5	6	7	8	9	10
$y(t)$	10,500	17,000	26,500	39,000	54,500	73,000	94,500	119,000	146,500	177,000
$y_r(t)$	9,926	18,800	28,453	36,427	50,788	73,188	88,813	122,204	147,983	175,676
$e(t)$	-0,575	1,800	1,953	-2,573	-3,712	0,188	-5,687	3,204	1,483	-1,324

TAB. 11.1: Tabela com a posição real $y(t)$, as medições do radar $y_r(t)$ e o erro $e(t)$ associado às 10 medições realizadas por um radar com $\sigma = 3m$ e $f = 1Hz$.

Agrupando os coeficientes $a_{i,n} = [t_{i,n}^2, t_{i,n}, 1]$, $1 < n < 10$ em **A** e as constantes em **b** obtemos

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 2 & 1 \\ 9 & 3 & 1 \\ 16 & 4 & 1 \\ 25 & 5 & 1 \\ 36 & 6 & 1 \\ 49 & 7 & 1 \\ 64 & 8 & 1 \\ 81 & 9 & 1 \\ 100 & 10 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 9,926 \\ 18,800 \\ 28,453 \\ 36,427 \\ 50,788 \\ 73,188 \\ 88,813 \\ 122,204 \\ 147,983 \\ 175,676 \end{bmatrix} \quad (11.7)$$

Em seguida, resolvendo por decomposição QR, obtemos as seguintes matrizes:

$$\mathbf{Q} = \begin{bmatrix} -0,006 & -0,181 & 0,765 \\ -0,025 & -0,312 & 0,425 \\ -0,057 & -0,395 & 0,156 \\ -0,101 & -0,429 & -0,043 \\ -0,157 & -0,413 & -0,170 \\ -0,226 & -0,349 & -0,227 \\ -0,308 & -0,236 & -0,213 \\ -0,402 & -0,073 & -0,128 \\ -0,509 & 0,138 & 0,028 \\ -0,629 & 0,398 & 0,255 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} -159,163 & -19,006 & -2,419 \\ 0 & -4,877 & -1,851 \\ 0 & 0 & 0,850 \end{bmatrix} \quad (11.8)$$

Finalmente, a solução é obtida resolvendo-se a EQ. 11.9

$$\mathbf{x}^* = \mathbf{R}^{-1} \mathbf{Q}^\top \mathbf{b} = \begin{bmatrix} 1,610 \\ 0,747 \\ 9,155 \end{bmatrix} = \begin{bmatrix} a \\ v_0 \\ y_0 \end{bmatrix} \quad (11.9)$$

Logo, com apenas 10 medições realizadas por meio de um radar com precisão $\sigma = 3m$, obtemos como estimativas a aceleração $a = 1,61m/s^2$, a velocidade inicial $v_0 = 0,75m/s$ e a posição inicial $y_0 = 9,16m$. Comparando com os dados originais

$$y(t) = \frac{1}{2}at^2 + v_0t + y_0 = \frac{1}{2}1,5t^2 + 2t + 7 \quad (11.10)$$

$$y_r(t) = \frac{1}{2}1,61t^2 + 0,75t + 9,16 \quad (11.11)$$

percebemos que a estimativa da aceleração ficou próxima do valor real, porém as estimativas da velocidade e posição iniciais não foram tão boas. Isso se deve a dois fatores: o tamanho do conjunto de medições ($n = 10$) e a precisão do radar ($\sigma = 3m$). A FIG. 11.1 apresenta a curva real, a curva ajustada e as medições realizadas para o EX. 11.1.

Com relação ao tamanho do conjunto, o mesmo exemplo foi executado com 100 medições, obtendo-se o seguinte resultado:

$$\mathbf{x}^* = \begin{bmatrix} a \\ v_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 1,499 \\ 2,045 \\ 6,810 \end{bmatrix} \quad (11.12)$$

Exemplo 11.1

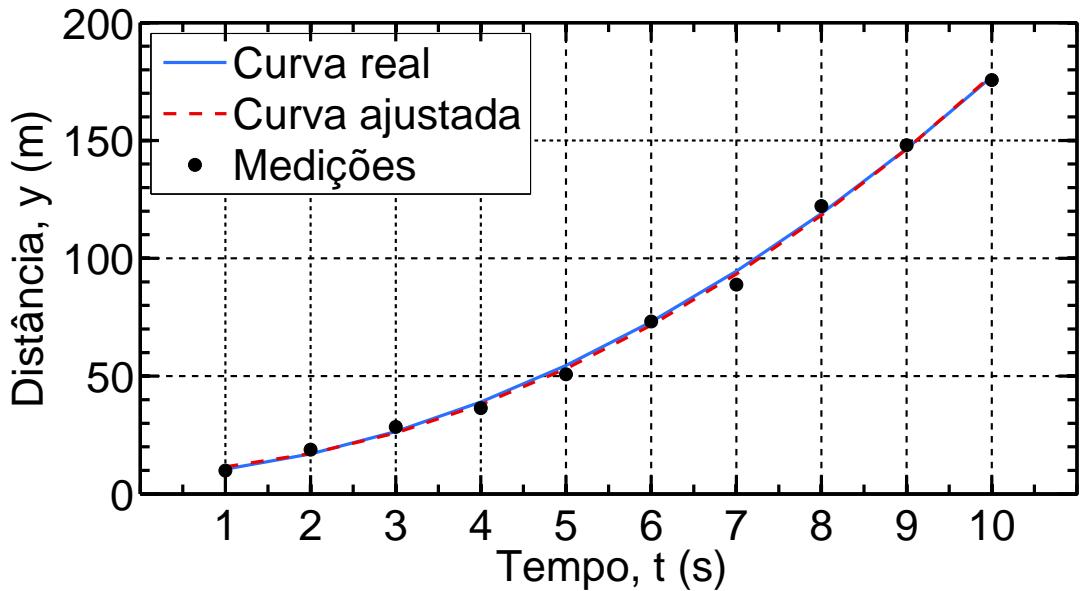


FIG. 11.1: Resultado da estimativa do EX. 11.1.

Podemos ver que o ajuste, ou melhor, a estimativa ficou bem melhor, pois o intervalo de medições considerado também foi maior. Esse tipo de situação é um exemplo simples para o conceito de suavização (*smoothing*). Na suavização todas as medições até o tempo t são levadas em consideração. O problema é que o tempo para o cálculo da estimativa também irá aumentar. Veremos na próxima seção como resolver essa questão.

No que tange à qualidade do sensor, suponha que agora disponhamos de outro radar, porém com precisão $\sigma = 0,5\text{m}$ e taxa de operação menor ($f = \frac{1}{3}\text{Hz}$). Como podemos levar essas condições em consideração? A solução é ponderar, dar um peso maior para as medições mais precisas e outro menor para as mais ruidosas. Abordaremos o assunto em seguida.

11.1.2 PROBLEMAS LINEARES PONDERADOS

Na seção anterior supomos que todas as medições possuíam a mesma influência. Isso nem sempre acontece. Aliás, é bastante comum nos problemas robóticos contarmos com sensores diversos cada um com suas características próprias. Dessas características, duas se destacam: a frequência de operação e a qualidade do sensor. A frequência informa o intervalo com que as medições são realizadas, sendo medido em Hz. A qualidade normalmente é dada por medidas estatísticas como a variância ou o desvio padrão. Dessa forma,

é possível adicionar o grau de influência de cada medição, ou mais especificamente, de cada sensor. Lembrando que no caso não ponderado a função objetivo era dada por

$$F(\mathbf{x}) = \frac{1}{2} \sum_i^n f_i(\mathbf{x})^2 = \frac{1}{2} \sum_i^n \|f_i\|_2^2 = \frac{1}{2} \mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) \quad (11.13)$$

onde $\|\cdot\|_2^2$ é o quadrado da distância euclidiana. Se as variâncias estão disponíveis podemos construir a matriz de covariância Σ_i para as medições \mathbf{y}_i . Nesse caso, a distância euclidiana é substituída pela distância de Mahalanobis $\|\cdot\|_{\Sigma_i}^2$, definida como $\|a - b\|_{\Sigma}^2 = (a - b)^T \Sigma^{-1} (a - b)$ (DE MAESSCHALCK, 2000). A função objetivo assume então a forma

$$\begin{aligned} F(\mathbf{x}) &= \frac{1}{2} \sum_i^n \|f_i\|_{\Sigma_i}^2 \\ &= \frac{1}{2} \mathbf{f}(\mathbf{x})^\top \Sigma^{-1} \mathbf{f}(\mathbf{x}) \\ &= \frac{1}{2} \mathbf{f}(\mathbf{x})^\top \Omega \mathbf{f}(\mathbf{x}) \end{aligned} \quad (11.14)$$

sendo Ω a matriz de informação, que é o inverso da matriz de covariância. A otimização passa a ser então

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} F(\mathbf{x}) = \arg \min_{\mathbf{x}} \frac{1}{2} \mathbf{f}(\mathbf{x})^\top \Omega \mathbf{f}(\mathbf{x}) \quad (11.15)$$

Adotando os mesmos passos realizados para obter a EQ. 11.4, o resultado é bastante semelhante:

$$\mathbf{A}^\top \Omega \mathbf{A} \mathbf{x}^* = \mathbf{A}^\top \Omega \mathbf{b} \quad (11.16)$$

A única diferença é a presença da matriz de informação Ω que simplesmente contém os recíprocos das variâncias de cada medição, funcionando dessa forma como uma matriz de pesos. Para compreendermos melhor a importância de se considerar as características dos sensores ou medições, vamos revisitar o EX. 11.1.

Exemplo 11.2. Suponha que além do primeiro sensor ($\sigma = 3m$, $f = 1Hz$) adquirimos um outro sensor com qualidade melhor, porém mais lento ($\sigma = 0,5m$, $f = \frac{1}{3}Hz$). Possivelmente em função de cálculos mais complexos para obter sua melhor precisão. Para simplificar, quando os dois sensores realizam uma medição num mesmo instante, descartamos a informação do primeiro sensor. A TAB. 11.2 apresenta o resultado das medições atualizado com os dados do novo sensor.

t [s]	1	2	3	4	5	6	7	8	9	10
y(t)	10,500	17,000	26,500	39,000	54,500	73,000	94,500	119,000	146,500	177,000
y_r(t)	9,926	18,800	26,211	36,427	50,788	73,327	88,813	122,204	147,167	175,676
e(t)	-0,575	1,800	-0,289	-2,573	-3,712	0,327	-5,687	3,204	0,667	-1,324

TAB. 11.2: Tabela com a posição real $y(t)$, as medições $y_r(t)$ e o erro $e(t)$ associado às 10 medições realizadas por dois radares (O primeiro com $\sigma = 3m$ e $f = 1Hz$ e o outro com $\sigma = 0,5m$ e $f = \frac{1}{3}Hz$).

Após realizarmos a otimização pelos mesmos procedimento já descritos obtemos:

$$\mathbf{x}^* = \begin{bmatrix} a \\ v0 \\ y0 \end{bmatrix} = \begin{bmatrix} 1,508 \\ 2,052 \\ 6,505 \end{bmatrix} \quad (11.17)$$

Podemos ver na FIG. 11.2 as medições, a curva real e a curva ajustada para o EX. 11.2. Repare que as curvas praticamente se confundem. O simples fato de termos mais um sensor com maior precisão já garantiu melhores resultados. Esse exemplo nos passa a ideia intuitiva sobre o que vem a ser a fusão de sensores: metodologia utilizada para a combinação de dados de diversos sensores com o objetivo de melhorar a qualidade das informações, de modo que ela seja, de algum modo, melhor do que se os sensores fossem utilizados individualmente (MITCHELL, 2012). Desse fato surgem duas importantes decisões de projeto: a escolha dos sensores e a definição da técnica de fusão de sensores. Os diversos sensores escolhidos, suas características e modelos serão abordados na seção 3.2. A técnica de fusão de sensores, dada a sua importância no desenvolvimento desta dissertação, será abordada no Capítulo 4. Contudo, apesar das melhorias apresentadas até o momento, consideramos somente os casos em que as funções objetivo eram lineares. Na prática, boa parte do modelos robóticos são não lineares. Veremos, em seguida, como tratar o problema.

11.1.3 PROBLEMAS NÃO-LINEARES

Diferentemente do que ocorre nos problemas lineares, no caso não-linear, as funções objetivo dependem de \mathbf{x} de modo não linear. Dessa forma, as funções f_i não podem ser escritas como uma combinação linear dos coeficientes de \mathbf{x} , conforme a EQ. 11.2. A maneira para resolver tais problemas encontra-se na linearização de tais funções em torno de um ponto

Exemplo 11.2

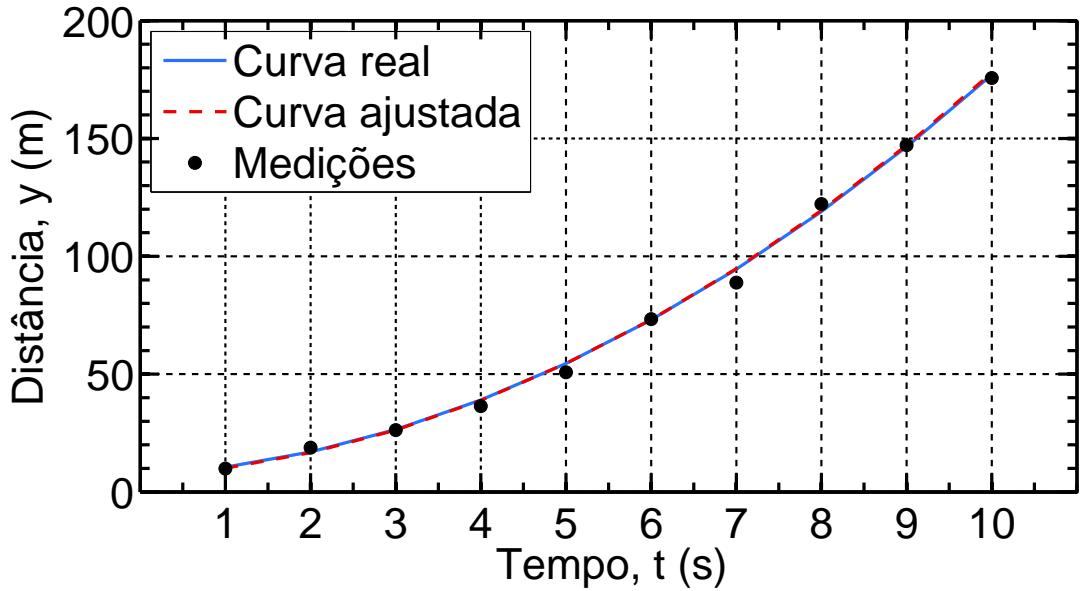


FIG. 11.2: Resultado da estimativa do EX. 11.2.

(suposição) inicial \mathbf{x}_0 e em seguida aplicar métodos iterativos⁷ com o objetivo de convergir ou aproximar-se da melhor solução \mathbf{x}^* (mínimo global). Na prática, esse resultado só é garantido para funções convexas⁸. No caso geral, dependendo da suposição inicial \mathbf{x}_0 , podemos encontrar diferentes mínimos locais. Lembrando que no caso das funções quadráticas temos

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} F(\mathbf{x}) = \arg \min_{\mathbf{x}} \frac{1}{2} \mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) \quad (11.18)$$

e aplicando a linearização pela expansão de Taylor de primeira ordem em torno do ponto \mathbf{x} :

⁷Dentre os diversos métodos iterativos temos: *Gradient descent*, método de Newton, Gauss-Newton, Levenberg-Marquardt e Powell's Dogleg. Detalhes sobre esses diversos métodos encontram-se em (MADSSEN, 2004).

⁸De maneira simplificada, são funções que possuem somente um mínimo. Pense, por exemplo, na função $y = x^2$, que é a nossa conhecida parábola. Seu único mínimo ocorre em $x = 0$. Agora pense na função periódica $y = \sin(x)$. Seus mínimos são infinitos e ocorrem em $x = \{\frac{3\pi}{2} + 2k\pi \mid k \in \mathbb{Z}\}$.

$$\begin{aligned}
F(\mathbf{x} + \Delta\mathbf{x}) &= \frac{1}{2}\mathbf{f}(\mathbf{x} + \mathbf{J}\Delta\mathbf{x})^\top \mathbf{f}(\mathbf{x} + \mathbf{J}\Delta\mathbf{x}) \\
&= \frac{1}{2}\mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) + \mathbf{f}(\mathbf{x})^\top \mathbf{J}\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^\top \mathbf{J}^\top \mathbf{J}\Delta\mathbf{x} \\
&= F(\mathbf{x}) + \mathbf{f}(\mathbf{x})^\top \mathbf{J}\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^\top \mathbf{J}^\top \mathbf{J}\Delta\mathbf{x}
\end{aligned} \tag{11.19}$$

onde \mathbf{J} é o Jacobiano de \mathbf{f} no ponto \mathbf{x} . Dada a aproximação realizada pela linearização, nós agora procuramos o $\Delta\mathbf{x}$ que minimiza a função. Logo, calculando a derivada primeira e igualando a zero:

$$\begin{aligned}
F'(\mathbf{x} + \Delta\mathbf{x}) &= 0 \\
\mathbf{f}(\mathbf{x})^\top \mathbf{J} + \mathbf{J}^\top \mathbf{J}\Delta\mathbf{x} &= 0 \\
\mathbf{J}^\top \mathbf{J}\Delta\mathbf{x} &= -\mathbf{J}^\top \mathbf{f}(\mathbf{x})
\end{aligned} \tag{11.20}$$

Essa derivação é conhecida como o método de Gauss-Newton e produz uma equação normal como a da EQ. 11.4, onde a matriz \mathbf{A} foi substituída pela matriz \mathbf{J} . Dessa forma, ela pode ser resolvida pelas ferramentas de decomposição já descritas.

11.1.4 PROBLEMAS NÃO-LINEARES PONDERADOS

No caso dos problemas não-lineares ponderados, a derivação é feita de forma análoga resultando em:

$$\mathbf{J}^\top \Omega \mathbf{J} \Delta\mathbf{x} = -\mathbf{J}^\top \Omega^\top \mathbf{f}(\mathbf{x}) \tag{11.21}$$