

Rodrigo Paz França

**Técnicas de Localização para um Veículo  
Aquático baseadas em Mapas Batimétricos e  
Filtro de Partículas**

**Porto Alegre - RS, Brasil**

**2016**

Rodrigo Paz França

# **Técnicas de Localização para um Veículo Aquático baseadas em Mapas Batimétricos e Filtro de Partículas**

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

Área de concentração: Sinais, Sistemas e Tecnologia da Informação

Linha de Pesquisa: Automação e Sistemas.

Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS

Faculdade de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Aurélio Tergolina Salton

Porto Alegre - RS, Brasil

2016



Pontifícia Universidade Católica do Rio Grande do Sul

FACULDADE DE ENGENHARIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

## TÉCNICAS DE LOCALIZAÇÃO PARA UM VEÍCULO AQUÁTICO BASEADAS EM MAPAS BATIMÉTRICOS E FILTRO DE PARTÍCULAS.

**CANDIDATO: RODRIGO PAZ FRANCA**

Esta Dissertação de Mestrado foi julgada para obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul.

**DR. AURELIO TERGOLINA SALTON - ORIENTADOR**

### BANCA EXAMINADORA

**DR. DIEGO ECKHARD - PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA-PPGEE,  
PÓS-GRADUAÇÃO E MATEMÁTICA APLICADA-PPGMAP - UFRGS**

**DR. JEFERSON VIEIRA FLORES - PÓS-GRADUAÇÃO EM ENGENHARIA  
ELÉTRICA-PPGEE - UFRGS**

**DR. ALEXANDRE ROSA FRANCO - DO PPGEE/FENG - PUCRS**

**PUCRS**

Campus Central

Av. Ipiranga, 6681 - Prédio 30 - Sala 103 - CEP: 90619-900

Telefone: (51) 3320.3540 - Fax: (51) 3320.3625

E-mail: engenharia.pg.eletrica@puers.br

*À Iana, e a minha família por todo o apoio, carinho e incentivo ao longo desta jornada.*

# Agradecimentos

Gostaria de agradecer primeiramente ao meu orientador, Prof. Aurélio Tergolina Salton pelo conhecimento, empenho e cooperação que foram essenciais para elaboração deste trabalho. Aos professores Prof. Jeferson Vieira Flores, e Prof. Alexandre Rosa Franco pelos conselhos e ajuda. Agradeço aos amigos e colegas do laboratório *EASE* pelo excelente convívio, amizade e companheirismo. Agradeço especialmente a Iana pela incansável dedicação, compreensão, carinho, amizade, companheirismo e incentivo durante todo o processo de desenvolvimento deste trabalho. Também agradeço a minha família pelo apoio e incentivo.

# Resumo

Este trabalho apresenta métodos de localização para veículos aquáticos baseados na fusão de sensores. Para realizar a fusão dos sensores foram utilizadas duas técnicas: o Filtro de Kalman Estendido (FKE) e o Filtro de Partículas (FP). O método do FKE desenvolvido, além de uma Unidade de Medidas Inercias (*IMU* do inglês-*Inertial Measurement Unit*) também necessita de um sensor GIB (*GPS Intelligent Buoys*). Com a finalidade de desenvolver um método de localização de baixo custo, o qual não necessite de sensores externos, foi desenvolvido a técnica de Localização Baseada em Terreno (LBT). Tal método consiste na fusão entre uma *IMU*, um sonar e um mapa batimétrico da região de navegação do veículo, por meio do FP. Esta técnica utiliza partículas para estimar a função de probabilidade da posição do veículo; entretanto, tem como limitação a baixa precisão em regiões de pouca variação de profundidade no mapa. A fim de contornar esse problema são apresentadas duas soluções, uma em *software* que utiliza um algoritmo de geração de trajetória (GT) que força o veículo a navegar por locais onde a variação de terreno é maior. Também foi desenvolvido uma solução que utiliza um sensor GIB como periférico, com propósito de melhorar o passo de correção do FP. A fim de validar os métodos foram realizadas simulação no *software MATLAB* utilizando um modelo matemático de um AUV (*Autonomous Underwater Vehicle*). Uma análise do custo computacional da técnica LBT foi realizada através da implementação do algoritmo na linguagem de programação “C”, onde foi embarcada em uma placa micro-controlada para medir a frequência de execução do método. Foram realizadas mil simulações para técnica de LBT e para LBT em conjunto com o algoritmo de GT com diferentes quantidades de partículas, com o intuito de analisar a capacidade de convergência do método de LBT e também o efeito do algoritmo de GT.

**Palavras-chaves:** Localização, Filtro de Partículas, Filtro de Kalman Estendido, Geração de Trajetória, *IMU*, Veículo Aquático, Localização Baseada em Terreno.

# Abstract

This work presents localization methods for underwater vehicles based on sensor fusion. To achieve the sensor fusion two techniques were used: Extended Kalman Filter (EKF) and Particle Filter (PF). The EKF method developed uses an Inertial Measurement Unit (IMU) and a GPS Intelligent Buoys (GIB) sensor. With the purpose of developing a low-cost location method which does not require external sensors, the Terrain Based Localization (TBL) technique was also developed. This method consists in the fusion between an IMU, a sonar and a bathymetric map of the navigation region through the PF. This technique utilizes particles to estimate the probability function of the vehicle position, however, this approach shows poor precision in regions of low depth variation. In order to solve this problem two solutions are presented, one in software, which uses a trajectory generation algorithm (TG) that limits the vehicle navigation to regions of the map with large depth variation; and another solution which uses a peripheral GIB sensor, with the purpose of improving the PF correction step. In order to validate the developed methods simulations in the *MATLAB* software were made utilizing an AUV (Autonomous Underwater Vehicle) mathematical model. An analysis of the computational cost of the LBT technique was performed, through the implementation of the algorithm in the “C” programming language, where it was embedded on a micro-controlled board to measure the method execution time. In order to analyze the convergence capacity of the TBL and the TG algorithm effect thousand simulations with different quantities of particles were performed for TBL technique and TBL together with the GT algorithm.

**Key-words:** Localization, Particle Filter, Extended Kalman Filter, Trajectory Generation, IMU (Inertial Measurement Unit), Underwater Vehicle, Terrain Based Localization.

# Lista de ilustrações

Figura 1 – Modelagem da localização de um veículo aquático em um plano $2D$ , onde $x$ , $y$ e $\psi$ representam a <i>pose</i> do sistema no eixo de coordenadas global e $\nu_1$ , $\nu_2$ e $\nu_3$ são respectivamente as velocidades lineares e angular no eixo de coordenadas do veículo . . . . .	23
Figura 2 – Estrutura do mosaico . . . . .	26
Figura 3 – Sistema com boias que utilizam GPS (a) sistema GIB (b)boias (c) pinger . . . . .	27
Figura 4 – Sistema acústico (a) <i>Long Base Line</i> (b) <i>Short Base Line</i> (c) <i>Ultra-Short base line</i> . . . . .	28
Figura 5 – Mapa batimétrico do <i>Mono Lake</i> califórnia . . . . .	30
Figura 6 – Localização baseada no terreno . . . . .	30
Figura 7 – Algoritmo do Filtro de Kalman. . . . .	32
Figura 8 – Algoritmo do Filtro de Kalman Estendido. . . . .	34
Figura 9 – Algoritmo do Filtro de Partículas. . . . .	36
Figura 10 – Figura mostra a ideia por trás do método de <i>Resample</i> no FP: (a) O objetivo é aproximar a função densidade $f$ . (b) Como não é possível gerar amostras diretamente de $f$ utiliza-se uma função densidade diferente $g$ , e as amostras geradas (partículas $\mathbf{x}^M$ ) a partir de $g$ são representadas na parte de baixo do diagrama. (c) Para obter uma amostra de $f$ e necessário atribuir o peso de importância $f(\mathbf{x})/g(\mathbf{x})$ para cada partícula $\mathbf{x}^M$ . No FP $f$ corresponde a função densidade do estado real $\mathbf{x}$ e $g$ ao estimado $\hat{\mathbf{x}}$ . . . . .	38
Figura 11 – Figura que mostra o veículo LAUV (parte de baixo) . . . . .	40
Figura 12 – A) Mapa batimétrico sintético. B) Mapa batimétrico real. . . . .	41
Figura 13 – Representação da modelagem do sonar, a linha marrom representa o solo submarino e a linha azul a superfície da água. A seta vermelha representa a distancia medida pelo sonar $z(x, y)$ , a seta preta mostra a profundida $d$ do veículo, e a seta roxa descreve o valor de profundidade obtido pelo mapa $M_{mapa}(x, y)$ . . . . .	42
Figura 14 – Representação do sistema GIB modelado, onde o triangulo vermelho representa a boia (GIB), a seta da cor preta a distancia $r^j$ medida, a linha azul representa a superfície da água e a linha marrom o solo aquático . . . . .	43
Figura 15 – Diagrama de blocos que representa a estrutura do FKE desenvolvido, onde $\nu(3)$ , $\nu(1)$ e $\nu(2)$ são respectivamente a velocidade angular e as acelerações lineares nos eixos $x$ e $y$ do modelo do veículo. . . . .	44



Figura 16 – Algoritmo de localização utilizando o Filtro de Kalman Estendido. . . .	47
Figura 17 – Representação do veículo nos dois eixos de coordenadas. . . . .	48
Figura 18 – Algoritmo de localização utilizando o Filtro de Kalman Estendido com o sensor virtual. . . . .	50
Figura 19 – Trajetórias utilizadas para obter os resultados utilizando a técnica de localização baseada no FKE. A) primeira trajetória. B) Segunda trajetória	51
Figura 20 – Localização da boia utilizada para o método de localização baseado no FKE com uma boia. . . . .	52
Figura 21 – Resultado do FKE para a primeira trajetória utilizando uma boia . . .	53
Figura 22 – Erro da estimação do filtro para a primeira trajetória . . . . .	53
Figura 23 – Resultado do FKE para a segunda trajetória utilizando uma boia . . .	54
Figura 24 – Erro da estimação do filtro para a segunda trajetória . . . . .	54
Figura 25 – Mapa com as posições das duas boias utilizadas . . . . .	55
Figura 26 – Resultado do FKE para a primeira trajetória utilizando duas boias . .	56
Figura 27 – Erro da estimação do FKE para a primeira trajetória utilizando duas boias . . . . .	56
Figura 28 – Resultado do FKE para a segunda trajetória utilizando duas boia . . .	57
Figura 29 – Erro da estimação do FKE para a segunda trajetória utilizando duas boias . . . . .	57
Figura 30 – Mapa com as posições das boias . . . . .	58
Figura 31 – Resultado do FKE para a primeira trajetória utilizando três boias . . .	59
Figura 32 – Erro da estimação obtido utilizando o FKE com três boias para pri- meira trajetória . . . . .	59
Figura 33 – Resultado do FKE para a segunda trajetória utilizando três boias . . .	60
Figura 34 – Erro da estimação obtido com o FKE para a segunda trajetória utili- zando três boias . . . . .	60
Figura 35 – Resultado do FKE para primeira trajetória utilizando três boias e com sensor virtual . . . . .	61
Figura 36 – Erro da estimação obtido com o FKE para primeira trajetória utili- zando três boias e com sensor virtual . . . . .	62
Figura 37 – Resultado do FKE para segunda trajetória utilizando três boias e com sensor virtual . . . . .	63
Figura 38 – Erro da estimação obtido com o FKE para a segunda trajetória utili- zando três boias e com sensor virtual . . . . .	64
Figura 39 – Primeiro passo da predição. A) Veículo na posição inicial sem a iniciali- zação das partículas. B) Inicialização da partículas em torno da posição inicial do veículo . . . . .	67
Figura 40 – Ilustração do modelo RTR: Rotação inicial $\delta_{r1}$ , translação $\delta_{t1}$ e rotação final $\delta_{r2}$ . . . . .	68

Figura 41 – Algoritmo do modelo RTR. . . . .	69
Figura 42 – Segundo passo da predição. A) movimentação das partículas pelo RTR B) as partículas continuam a se mover governadas pelo RTR até ocorrer a correção . . . . .	69
Figura 43 – Efeitos dos $\alpha$ A) Todos os $\alpha = 0$ . B) Todos os $\alpha = 0$ exceto o $\alpha_3 = 100$ . C) Todos os $\alpha = 0$ exceto o $\alpha_2 = 7$ . D) Mais de um $\alpha$ diferente de zero.	70
Figura 44 – Algoritmo de atribuição dos pesos $w$ para cada partícula. . . . .	71
Figura 45 – Algoritmo de seleção das melhores partículas. . . . .	73
Figura 46 – Diagrama de blocos da técnica de localização baseado no FP utilizando boias . . . . .	74
Figura 47 – Diagrama do Filtro de Partículas baseado em terreno. . . . .	74
Figura 48 – Algoritmo de atribuição dos pesos $w$ para cada partícula. . . . .	75
Figura 49 – Exemplo do passo de correção do método LBT A) Partículas se dis- persando governadas pelo modelo RTR. B) Ação do passo de correção, onde as partículas com baixa probabilidade foram eliminadas. . . . .	76
Figura 50 – Problema da Localização Baseada em terreno, as partículas não con- seguem se diferenciar em regiões com profundidade igual. A linha roxa mostra a trajetória real do veículo. . . . .	77
Figura 51 – Algoritmo do calculo da variação do terreno. . . . .	78
Figura 52 – Algoritmo da <i>Payoff Function</i> . . . . .	79
Figura 53 – Resultado obtido para a primeira trajetória com a técnica de LBT, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas . . . . .	82
Figura 54 – Resultado obtido para a segunda trajetória com a técnica de LBT, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas . . . . .	83
Figura 55 – Resultado obtido para a primeira trajetória utilizando o Filtro de Par- tículas com duas boias, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória esti- mada pelo filtro através da média de todas as partículas . . . . .	84
Figura 56 – Resultado obtido para a segunda trajetória utilizando o Filtro de Par- tículas com duas boias, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória esti- mada pelo filtro através da média de todas as partículas . . . . .	85

Figura 57 – Resultado obtido para a primeira trajetória utilizando o Filtro de Partículas com três boias, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas . . . . .	86
Figura 58 – Resultado obtido para a segunda trajetória utilizando o Filtro de Partículas com três boias, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas . . . . .	87
Figura 59 – Resultado obtido para a primeira trajetória com a técnica de LBT em conjunto com o uma boia, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas . . . . .	88
Figura 60 – Resultado obtido para a segunda trajetória com a técnica de LBT em conjunto com uma boia, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas . . . . .	89
Figura 61 – Resultado obtido para a primeira trajetória com a técnica de LBT em conjunto com o algoritmo de geração de trajetória, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas . . . . .	91
Figura 62 – Resultado obtido para a segunda trajetória com a técnica de LBT em conjunto com o algoritmo de geração de trajetória, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas . . . . .	92
Figura 63 – Resultado obtido para a primeira trajetória com a técnica de LBT em conjunto com o algoritmo de geração de trajetória, e o sensor GIB, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas . . . . .	93
Figura 64 – Resultado obtido para a segunda trajetória com a técnica de LBT em conjunto com o algoritmo de geração de trajetória, e o sensor GIB, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas . . . . .	94
Figura 65 – Resultados obtidos com mil simulações para 200 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.	96

Figura 66 – Resultados obtidos com mil simulações para 500 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.	97
Figura 67 – Resultados obtidos com mil simulações para 1000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.	97
Figura 68 – Resultados obtidos com mil simulações para 2000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.	98
Figura 69 – Resultados obtidos com mil simulações para 3000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.	98
Figura 70 – Resultados obtidos com mil simulações para 4000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.	99
Figura 71 – Resultados obtidos com mil simulações para 5000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.	99
Figura 72 – Resultados obtidos na estimação do ângulo $\psi$ com mil simulações para 200 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória. . . . .	100
Figura 73 – Resultados obtidos na estimação do ângulo $\psi$ com mil simulações para 500 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória. . . . .	100
Figura 74 – Resultados obtidos na estimação do ângulo $\psi$ com mil simulações para 1000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória. . . . .	101
Figura 75 – Resultados obtidos na estimação do ângulo $\psi$ com mil simulações para 2000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória. . . . .	101
Figura 76 – Resultados obtidos na estimação do ângulo $\psi$ com mil simulações para 3000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória. . . . .	102
Figura 77 – Resultados obtidos na estimação do ângulo $\psi$ com mil simulações para 4000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória. . . . .	102
Figura 78 – Resultados obtidos na estimação do ângulo $\psi$ com mil simulações para 5000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória. . . . .	103

# Lista de tabelas

Tabela 1 – Tabela com os valores de media ( $\mu$ ) e covariância ( $\sigma$ ) utilizados nos ruídos dos sensores . . . . .	51
Tabela 2 – Tabela com o erro de estimação das trajetória para o método de localização baseado no FKE. . . . .	65
Tabela 3 – Tabela com o erro médio de estimação do ângulo $\psi$ para o método de localização baseado no FKE. . . . .	65
Tabela 4 – Tabela com os valores de media ( $\mu$ ) e covariância ( $\sigma$ ) utilizados nos ruídos dos sensores . . . . .	81
Tabela 5 – Tabela com os parâmetros utilizado no Filtro de Partículas . . . . .	81
Tabela 6 – Tabela com os parâmetros utilizado no algoritmo de geração de trajetória. Linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas . . . . .	90
Tabela 7 – Tabela com o erro média euclidiano da trajetória estimada utilizando os método de localização baseados no Filtro de Partículas . . . . .	95
Tabela 8 – Tabela com o erro médio do ângulo $\psi$ estimado utilizando os método de localização baseados no Filtro de Partículas . . . . .	95
Tabela 9 – Especificações do computador utilizado para computar os tempos de execução do FP . . . . .	103
Tabela 10 – Tabela com os tempos de execução do Filtro de Partículas com diferentes quantidades de partículas . . . . .	104
Tabela 11 – Tabela com o tempo de execução de cada <i>loop</i> do Filtro de Partículas no processador . . . . .	105

# Lista de abreviaturas e siglas

<i>FK</i>	Filtro de Kalman
<i>FKE</i>	Filtro de Kalman Estendido
<i>FP</i>	Filtro de Partículas
<i>LBT</i>	Localização Baseada em Terreno
<i>IMU</i>	<i>Inertial Measurement Unit</i>
<i>GIB</i>	<i>GPS Intelligent Buoys</i>
<i>LBL</i>	<i>Long Base Line</i>
<i>SBL</i>	<i>Short Base Line</i>
<i>USBL</i>	<i>Ultra Short Base Line</i>
<i>GT</i>	Técnica de Geração de Trajetória
<i>pose</i>	Estados que determinam a localização do veículo
<i>GPS</i>	<i>Global Positioning System</i>
<i>DR</i>	<i>Dead Reckoning</i>

# Lista de símbolos

$\mathbf{x}$	Estados do sistema
$\dot{\mathbf{x}}$	Derivada dos estados do sistema
$\bar{\mathbf{x}}$	Estados do sistema estimados a priori
$\hat{\mathbf{x}}$	Estados do sistema estimado a posteriori
$\eta$	Vetor de posição e orientação do modelo do veículo
$x$	Posição $x$ global do veículo
$y$	Posição $y$ global do veículo
$z$	Posição $z$ global do veículo
$\psi$	Ângulo $\psi$ global do veículo
$\phi$	Ângulo $\phi$ global do veículo
$\theta$	Ângulo $\theta$ global do veículo
$I_i$	Matriz identidade de tamanho $i$
$\dot{\nu}$	Vetor de velocidades do modelo do veículo
$\mathbf{u}$	Velocidade linear no eixo $x$ do modelo do veículo
$\mathbf{v}$	Velocidade linear no eixo $y$ do modelo do veículo
$\mathbf{w}$	Velocidade linear no eixo $z$ do modelo do veículo
$\mathbf{p}$	Velocidade angular referente ao ângulo $\phi$
$\mathbf{q}$	Velocidade angular referente ao ângulo $\theta$
$\mathbf{r}$	Velocidade angular referente ao ângulo $\psi$
$J(\psi_k)$	Matriz de rotação
$M$	Matriz de massa inercial do sistema
$D(\nu_k)$	Matriz de amortecimento

$\tau_k$	Vetor de forças de propulsão do modelo referente ao eixo de coordenadas do veículo
$\mathcal{N}(\mu, \sigma)$	Ruído branco com média $\mu$ e covariância $\sigma$
$\varepsilon_{acc}$	Ruído referente as medições do acelerômetro
$\varepsilon_{gy}$	Ruído referente as medições do giroscópio
$\varepsilon_z$	Ruído referente as medições do sonar
$\varepsilon_r$	Ruído referente as medições da boia
$\mu_{acc}$	Média do ruído referente as medições do acelerômetro
$\mu_{gy}$	Média do ruído referente as medições do giroscópio
$\mu_r$	Média do ruído referente as medições da boia
$\mu_z$	Média do ruído referente as medições do sonar
$\sigma_{acc}$	Covariância do ruído referente as medições do acelerômetro
$\sigma_{gy}$	Covariância do ruído referente as medições do giroscópio
$\sigma_r$	Covariância do ruído referente as medições da boia
$\sigma_z$	Covariância do ruído referente as medições do sonar
$M_{mapa}$	Matriz que representa o mapa batimétrico
$d$	Profundidade que o veículo está navegando
$z(x, y)$	Medição de profundidade do sonar
$r^j$	Medição da distancia do veículo até a boia
$\mathbf{m}_x$	Posição $x$ da boia
$\mathbf{m}_y$	Posição $y$ da boia
$\dot{\nu}(1)$	Velocidade linear no eixo $x$ do modelo do veículo
$\dot{\nu}(2)$	Velocidade linear no eixo $y$ do modelo do veículo
$\nu(3)$	Velocidade angular (ângulo $\psi$ ) do modelo do veículo
$a_{\mathbf{u}}$	Medida da aceleração no eixo $x$ oriunda do acelerômetro
$a_{\mathbf{v}}$	Medida da aceleração no eixo $yy$ oriunda do acelerômetro



$a_\psi$	Velocidade angular medida pelo giroscópio
$\Delta_x$	Integração das medidas no eixo $x$ do acelerômetro entre dois passos
$\Delta_y$	Integração das medidas no eixo $y$ do acelerômetro entre dois passos
$\Delta_\psi$	Integração das medidas do ângulo $\psi$ do giroscópio entre dois passos
$p_k$	<i>pose</i> do sistema
$\bar{p}_k$	<i>pose</i> do sistema estimada a priori
$\hat{p}_k$	<i>pose</i> do sistema estimada a posteriori
$\mathbf{P}(i)$	Probabilidade associada ao evento $i$
$\chi_k^M$	Grupo de partículas
$\delta_{r1}$	Primeira rotação do modelo RTR
$\delta_{t1}$	Primeira translação do modelo RTR
$\delta_{r2}$	Segunda rotação do modelo RTR
$\mu$	Média de um número aleatório
$\sigma^2$	Covariância de um número aleatório
$\alpha_1, \alpha_2, \alpha_3, \alpha_4$	Parâmetros do modelo RTR que determinam a variabilidade das partículas
$w_k^M$	Peso de importância final das partículas
$wr_k^M$	Peso de importância das partículas calculado com base nas medições das boias
$wp_k^M$	Peso de importância das partículas calculado com base nas medições do sonar
$\hat{r}^i$	Medição da distância virtual da boia até a partícula
$\beta$	Limiar de seleção das melhores partículas
$\hat{z}_k^M$	Medição virtual do sonar para cada partícula
$f(\mathbf{x}, u)$	Modelo de previsão do FKE
$\tilde{\mathbf{x}}$	Estimativa final dos estados do sistema dado pela média de todas as partículas

$P_k$	Matriz de covariância do FKE
$F_k$	Jacobiana do modelo de predição do Filtro de Kalman Estendido
$T_s$	Tempo de amostragem
$R_k$	Matriz de covariância do modelo de predição do Filtro de Kalman Estendido
$h_k^i$	Modelo de correção do FKE
$H_k^i$	Jacobiana do modelo de correção do FKE
$K_k$	Ganho do FKE
$Q_k$	Matriz de covariância do modelo de correção do Filtro de Kalman Estendido
$Z_k^i$	Modelo do sensor real utilizado no FKE
$R(\psi)$	Matriz de rotação
$V_x$	Vetor de velocidade global do eixo $x$
$V_y$	Vetor de velocidade global do eixo $y$
$Z_V$	Medição do sensor virtual
$\nabla v_0$	Taxa de movimento do sistema no eixo global
$\delta$	Ação ótima gerada pelo algoritmo de GT
$\pi_{pol}$	Melhor ação para cada estado
$V$	Função <i>value</i> que representa o <i>comulative payoff</i>
$\gamma$	Fator de desconto que garante convergência do algoritmo de GT
$R_{pay}(x, y)$	Função de <i>local payoff</i> que determina o custo de movimento para cada posição do mapa
$M_I$	Matriz normalizada que representa a variação do terreno
$F_x$	Gradiente do mapa calculado para direção $x$
$F_y$	Gradiente do mapa calculado para direção $y$
$x_g$	Posição $x$ de chegada do veículo
$y_g$	Posição $y$ de chegada do veículo

$\lambda_g$	Custo para se movimentar para posição de chegada
$\lambda_c$	Custo para se movimentar para qualquer posição do mapa
$\lambda_M$	Custo da direção de movimento
$\Upsilon$	Limiar que representa a importância do terreno

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>21</b>
1.1	Definição do Problema	23
1.2	Proposta	24
1.2.1	Objetivos	24
1.3	Organização do Texto	25
<b>2</b>	<b>PRELIMINARES</b>	<b>26</b>
2.1	Localização de Veículos Aquáticos	26
2.2	Filtro de Kalman	31
2.3	Filtro de Kalman Estendido (FKE)	32
2.4	Filtro de Partículas	34
2.4.1	Seleção das Melhores Partículas	37
2.5	Modelos Matemáticos	39
2.5.1	Modelo Matemático do Veículo Aquático	39
2.5.2	Modelo Matemático dos Sensores	41
<b>3</b>	<b>LOCALIZAÇÃO BASEADA NO FILTRO DE KALMAN ESTENDIDO</b>	<b>44</b>
3.1	Implementação do Filtro de Kalman Estendido	44
3.1.1	Predição	45
3.1.2	Correção	46
3.2	Correção utilizando um Sensor Virtual	47
3.3	Resultados	51
3.3.1	Resultados obtidos com uma Boia	52
3.3.2	Resultados obtidos com duas Boia	55
3.3.3	Resultados obtidos com três Boia	58
3.3.4	Resultados obtido com três boia e o sensor virtual	61
3.4	Análise de Desempenho	64
3.5	Comentários Finais	66
<b>4</b>	<b>LOCALIZAÇÃO BASEADA NO FILTRO DE PARTÍCULAS</b>	<b>67</b>
4.1	Predição	67
4.2	Correção Baseada em Boias	70
4.3	Localização Baseada em Terreno	74
4.4	Deficiência da Localização Baseada em terreno	76
4.5	Método de Geração de Trajetória	77
4.6	Resultados	80

4.6.1	Localização Baseada em Terreno (LBT) . . . . .	82
4.6.2	Localização Utilizando Boias . . . . .	83
4.6.3	LBT em Conjunto com Boias . . . . .	87
4.6.4	LBT em Conjunto com a Técnica de Geração de Trajetória . . . . .	89
4.6.5	LBT em conjunto com a Técnica de Geração de Trajetória e com Boias . . . . .	92
<b>4.7</b>	<b>Análise de Desempenho . . . . .</b>	<b>94</b>
<b>4.8</b>	<b>Análise de Convergência . . . . .</b>	<b>96</b>
<b>4.9</b>	<b>Análise de Custo Computacional . . . . .</b>	<b>103</b>
<b>4.10</b>	<b>Comentários Finais . . . . .</b>	<b>105</b>
<b>5</b>	<b>CONCLUSÕES E PERSPECTIVAS . . . . .</b>	<b>107</b>
<b>5.1</b>	<b>Perspectivas para Trabalhos Futuros . . . . .</b>	<b>108</b>
	<b>Referências . . . . .</b>	<b>109</b>

# 1 Introdução

O Brasil possui uma zona econômica chamada de “Amazônia Azul” que possui uma área oceânica de aproximadamente 3,6 milhões de  $km^2$ , e abrange mais de 17 estados. Tal zona possui muitas riquezas e uma grande importância estratégica para o país, sendo necessário conhecê-la e também garantir a sua proteção (AMAZÔNIA... , 2016). Outro fator importante é que em 2006 foi descoberto a reserva de petróleo chamada de “Pré-Sal” em águas ultraprofundas no litoral brasileiro, onde em 2014 atingiu uma produção de 492 mil barris por dia (PRÉ-SAL, 2016). Desta forma, o desenvolvimento de máquinas capazes de operar nesse terreno, a exemplo de submarinos, *AUVs* (*autonomous underwater vehicle*) se faz necessário. No entanto, existem alguns problemas que precisam ser resolvidos antes que tais veículos possam ser amplamente utilizados.

Uma das principais dificuldades em desenvolver veículos subaquáticos está relacionado a seu sistema de localização, porque não é possível utilizar diretamente o *GPS* (*Global Positioning System*) devido a atenuação do sinal embaixo da água (HYAKUDOME, 2011). Desta forma, é preciso utilizar outros sensores como câmeras, sensores acústicos, sensores inerciais, etc. Para solucionar esse problema técnicas baseadas em visão foram desenvolvidas, as quais utilizam câmeras presas ao veículo para detectar objetos (*landmarks*) e assim determinar a sua localização (ZHANG; MILIOS; GU, 2004). Contudo esses métodos que utilizam sensores ópticos possuem algumas limitações como o curto alcance das câmeras e a sua suscetibilidade a dispersão e insuficiência de luz (PAULL et al., 2014).

Como uma alternativa ao uso de câmeras existem os sensores acústicos, como *LBL* (*Long Base Line*), *SBL* (*Short Base Line*), *USBL* (*Ultra Short Base Line*) (VICKERY, 1998). Esses dispositivos fazem uso de transponders que são posicionados ao redor do veículo, formando assim um perímetro e, através do retorno de um sinal acústico, localizam o veículo. Esses sensores podem ser fixados em um navio que fica acompanhando o veículo, como é o caso do *SBL* e *USBL*, já o *LBL* é posicionado no solo submarino. O sistema *GIB* (*GPS Intelligent Buoys*) apresentado por YOUNGBERG, 1992 é outro sensor acústico composto por boias que ficam posicionadas na superfície da água (THOMAS, 1998). Entretanto esses dispositivos possuem algumas limitações como: o alto custo de um navio acompanhar o veículo durante o trajeto, o limite de alcance do sinal e também a necessidade de posicionar as boias e os transponders em todo o percurso.

O uso de sensores inerciais como a *IMU* (*Inertial Measurements Unit*) tem sido explorado para determinar a localização de um veículo aquático. Com esses sensores é possível medir a aceleração linear e angular de um corpo, e através da integração desses

dados pode-se determinar a posição e orientação do sistema, e esse método é chamado de *Dead Reckoning* (*DR*). Porém essa abordagem é suscetível a erros de *drift* que deterioram a estimação da posição (STUTTERS et al., 2008). Uma solução para esse problema é fazer com que o veículo suba até a superfície periodicamente para corrigir a sua posição com o sinal de *GPS*, mas isso se torna inviável em missões com grandes profundidades.

Por esses motivos pode-se dizer que desenvolver um sistema de localização robusto e preciso para um veículo aquático utilizando apenas um sensor é uma tarefa muito complicada. Em vez disso, em termos práticos, pode-se afirmar que a fusão de dados a partir de vários sensores é essencial para uma boa estimativa. A técnica mais conhecida utilizada para realizar a fusão de sensores é o Filtro de Kalman (FK) (KALMAN, 1960), o qual tem sido amplamente utilizado pela indústria para estimar estados, como por exemplo, aplicação na estimação da corrente de armadura de motores CC, na estimação de harmônicas em sistemas de potência (ZADEH; GHOSH; LEDWICH, 2010), para localização de robôs (WON et al., 2010) e em diagnósticos de falhas (ZHANG et al., 2009). Mas o FK pode ser aplicado apenas a sistemas lineares, sendo que a maioria dos sistemas reais são não-lineares. Para isso existe o Filtro de Kalman Estendido (FKE) (JULIER; UHLMANN, 1997). Entretanto o FKE lineariza e aproxima o sistema não-linear por meio da expansão por Série de Taylor, o que pode ocasionar algumas imperfeições na estimação dos estados (NEMRA; AOUF, 2010). As técnicas de localização para veículos aquáticos desenvolvidas utilizando o FKE empregam sensores como câmeras e sensores acústicos, o que pode elevar o custo de implementar o sistema na prática.

Outro filtro que vem ganhando espaço na área de estimação de estados é o Filtro de Partículas (FP) (GORDON; SALMOND; SMITH, 1993). Isso porque essa técnica não realiza linearizações no modelo e consegue adaptar-se a qualquer tipo de função probabilística de incerteza, em razão de não utilizar parâmetros como média e variância para descrever a função de probabilidade como faz o FKE. Para solucionar o problema de estimação o FP utiliza amostras aleatórias (“partículas”) que se adaptam a natureza da incerteza da melhor maneira. Além disso o FP permite desenvolver métodos de localização para veículos aquáticos que não dependem de sensores externos como LBL, USBL, GIB e entre outros, permitindo diminuir o custo de implementação. Um exemplo disso é o método de Localização Baseada em Terreno (*LBT*), o qual utiliza apenas sensores inercias, um sonar e um mapa batimétrico (mapa com informações de profundidade) do local onde o veículo irá navegar (KARLSSON; GUSFAFSSON; KARLSSON, 2003). Entretanto essa técnica pode resultar em estimativas erradas, em casos onde o terreno não possua informação suficiente para o FP (FRANCA et al., 2015).

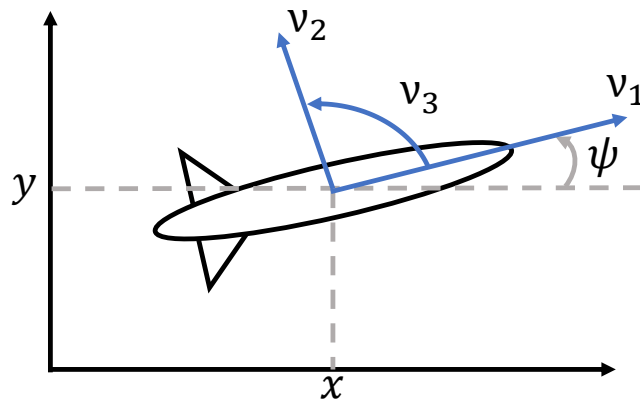
## 1.1 Definição do Problema

A localização de um robô móvel pode ser definida por meio de sua cinemática que é representada por seis variáveis, as três coordenadas do plano cartesiano ( $x, y$  e  $z$ ) e os três ângulos de Euler ( $\Phi$ -*roll*,  $\theta$ -*pitch* e  $\psi$ -*yaw*) relativo a um sistema de coordenadas externo. Esse conceito pode ser simplificado de maneira que a posição de um veículo em um plano (2D) pode ser definida através das duas variáveis de posição  $x$  e  $y$ , e pelo ângulo de orientação  $\psi$  (*yaw*). Desse modo essas três grandezas determinam a chamada pose  $p_k$  do sistema, que no instante de tempo  $k$  é descrita por :

$$p_k = \begin{bmatrix} x_k \\ y_k \\ \psi_k \end{bmatrix}. \quad (1.1)$$

Logo, é possível expandir esse conceito e definir a localização de um veículo aquático em um plano através de sua *pose* como é apresentado na Figura 1.

Figura 1 – Modelagem da localização de um veículo aquático em um plano  $2D$ , onde  $x$ ,  $y$  e  $\psi$  representam a *pose* do sistema no eixo de coordenadas global e  $\nu_1$ ,  $\nu_2$  e  $\nu_3$  são respectivamente as velocidades lineares e angular no eixo de coordenadas do veículo .



Fonte: O autor (2015)

Mas como um veículo se movimenta por todo o ambiente não existe uma maneira direta de definir instantaneamente a sua posição. Ao invés disso é necessário integrar o seu movimento ao longo do tempo, entretanto isso gera algumas incertezas, por isso pode-se dizer que determinar a localização de um veículo é uma tarefa desafiadora (SIEGWART; NOURBAKHS; SCARAMUZZA, 2011). Assim para realizar essa tarefa deve-se supor que dois tipos de sensores estão disponíveis, um sensor que descreva o deslocamento do sistema, e outro que forneça informação sobre o ambiente que o veículo se encontra. No



entanto, quando se trata do problema de localizar um veículo subaquático é necessário levar em consideração que não é possível utilizar alguns sensores como o GPS, devido que seu sinal é atenuado embaixo da água. Dessa forma deve-se utilizar outros sensores como câmeras, sensores acústicos, sensores inerciais, e entre outros.

Assim neste trabalho é realizado a fusão de dados de dois tipos de sensores para estimar a pose de um veículo aquático. Para isso duas técnicas são utilizadas, uma baseada no Filtro de Partículas (FP) e a outra no Filtro de Kalman Estendido (FKE).

## 1.2 Proposta

Dentro desse cenário este trabalho tem como objetivo desenvolver métodos de localização para um veículo aquático. Foram desenvolvidas duas abordagens diferentes, uma baseada no Filtro de Kalman Estendido e outra baseada no Filtro de Partículas. O método que utiliza o FKE desenvolvido emprega uma *IMU*, e um sensor *GIB*. Para validar a técnica desenvolvida simulações no *software MATLAB* foram feitas. Para o sistema *GIB* foram utilizadas uma, duas e três boias para realizar as simulações, a fim de determinar o menor número de boias necessários, já que a qualidade da estimação desta abordagem está diretamente ligada ao número de boias.

Com o Filtro de partículas foram desenvolvidas duas abordagens diferentes uma que é a Localização Baseada em Terreno e a outra que utiliza o sensor *GIB*. A técnica de LBT foi implementada visando um baixo custo, a qual utiliza uma *IMU*, um sonar e um mapa batimétrico, entretanto apresentou uma deficiência que ocorre quando o veículo navega por regiões do mapa com pouca variação de terreno. Logo para solucionar esse problema um algoritmo de Geração de Trajetória foi desenvolvido, o qual força o veículo a navegar por regiões do mapa que possuem grandes variações de profundidade. O outro método utiliza uma *IMU* e um sensor *GIB* com uma, duas e três boias para analisar qual é a quantidade mínima para obter uma boa estimação. Para validar essas técnicas, simulações foram realizadas no *software MATLAB*. Devido que o FP utiliza números aleatória para realizar a estimação da localização do veículo foram realizadas mil simulações para grupos com diferentes quantidades de partículas para analisar a convergência e desempenho da técnica. Como o FP exige um grande custo computacional, o algoritmo do filtro foi implementado em linguagem de programação “C” e embarcado em um placa microcontrolada, para assim determinar os requisitos de hardware necessário para implementar essa técnica na prática.

### 1.2.1 Objetivos

Os objetivos específicos deste trabalho são os seguintes:

- Desenvolver um algoritmo de localização para um AUV baseado na fusão de sensores utilizando o Filtro de Kalman Estendido.
- Desenvolver um algoritmo de localização para um AUV baseado na fusão de sensores utilizando o Filtro de Partículas.
- Comparar os dois métodos desenvolvidos.
- Melhorar o desempenho do PF com um algoritmo de geração de trajetória .
- Implementar o método de localização desenvolvido utilizando o FP em linguagem de programação C, para embarcar em um placa microcontrolada. Para assim analisar os requisitos mínimos de hardware necessários para implementação prática da técnica.
- Determinar o desempenho do algoritmo de Geração de trajetória.
- Analisar a do método de LBT (Localização Baseada em Terreno).

### 1.3 Organização do Texto

Este trabalho está organizado da seguinte maneira. No Capítulo 2 são descritos os conceitos preliminares para entendimento dos métodos propostos. Esse capítulo apresenta alguns métodos de localização para veículos aquáticos, filtros lineares e não lineares de estimação de estados como, o Filtro de Partículas, Filtro de Kalman e Filtro de Kalman Estendido e os modelos matemáticos do veículo e dos sensores. A definição do problema abordado neste trabalho, a proposta e os objetivos são descritos no Capítulo 1. O método de localização baseado no Filtro de Kalman Estendido é descrito no Capítulo 3, e a técnica de localização baseada no Filtro de partícula é relatada no Capítulo 4. E por último são descritas a conclusão e perspectivas futuras no Capítulo 5

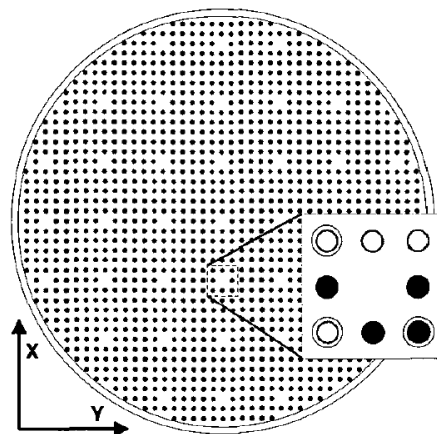
## 2 Preliminares

Neste capítulo são apresentados os conceitos necessários para entender o desenvolvimento deste trabalho. Primeiramente será tratado sobre os métodos de localização para veículos aquáticos já desenvolvidos. As duas técnicas de estimação não-lineares utilizadas são descritas posteriormente, e por último são apresentados os modelos matemáticos do veículo e dos sensores utilizados.

### 2.1 Localização de Veículos Aquáticos

Já foram desenvolvidos diversas técnicas para sanar o problema de localização de um AUV. Por exemplo, existem métodos que utilizam recursos do ambiente externo para determinar a sua posição e orientação, para isso fazem uso de câmeras. CARRERAS et al., 2003 apresenta um método para estimar a posição e a orientação do AUV utilizando uma câmera que fica presa na estrutura do veículo voltada para baixo e um mosaico posicionado no fundo de um tanque. O mosaico é formado por objetos com geometria circular e pelas cores: branca, que representa o fundo, e pretas e cinza, associadas aos pontos e círculos (Figura 2). Assim, a câmera capta os pontos presentes no mosaico e o algoritmo é capaz de determinar a posição do AUV. Dentro do contexto da ideia de utilizar um mosaico e uma câmera para determinar a localização de um veículo aquático, existem outros pesquisadores que desenvolveram seus próprios métodos como XU; NEGAHDARIPOUR, 1997 e GARCIA et al., 2001.

Figura 2 – Estrutura do mosaico

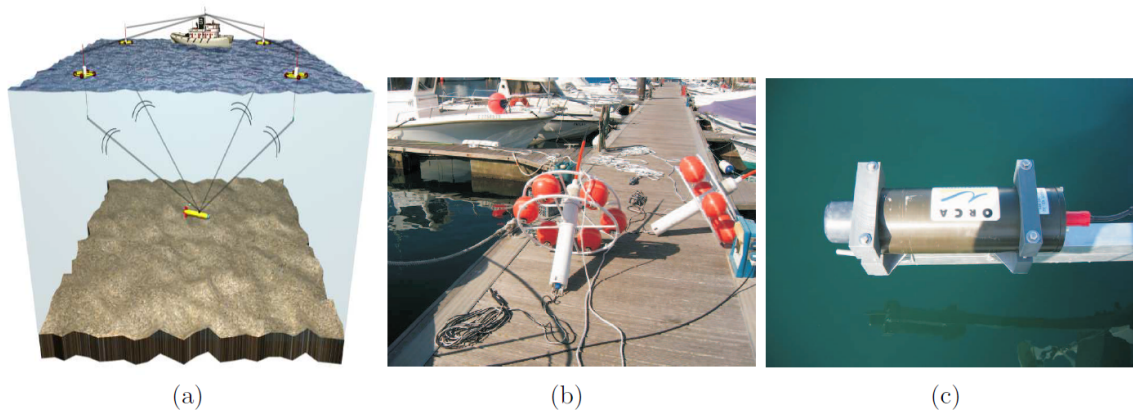


Fonte: CARRERAS et al., (2003).

Então, a ideia básica de utilizar câmeras para localização é poder reconhecer e identificar pontos de referencia que permitam determinar a localização do veículo. Mas essas técnicas possuem limitações como o alcance reduzido das câmeras em ambiente subaquático, a falta de iluminação e turvação, esses fatores se agravam a medida que a profundidade aumenta (KIM et al., 2014).

Existem técnicas que utilizam o GPS para estimar a posição, mas devido ao fato de que sinais eletromagnéticos são atenuados embaixo da água deve-se utilizar transmissores auxiliares. Por exemplo, existe uma tecnologia chamada GIB (Figura 3) (*GPS Inteligente Buoys*) (ALCOCER; OLIVEIRA; PASCOAL, 2006) que é constituída de quatro boias que ficam situadas na superfície e são equipadas com receptores DGPS e hidrofones. Cada um desses hidrofones capta um sinal de “ping” provenientes do alvo que está embaixo da água e grava os tempos de chegada. Assim as boias enviam essa informação para uma central que computa a posição do alvo que está embaixo da água. Com esses sensores também é possível determinar a profundidade do alvo, através do tempo de *delay* dos pulsos do *pinger*. Os tempos de chegada gravados pelos hidrofones podem ser convertidos em distancias entres as boias e o *pinger*, para isso deve-se conhecer a velocidade do som dentro da água. Com base nessa tecnologia CAITI et al., 2005 propõe um algoritmo de localização para um AUV.

Figura 3 – Sistema com boias que utilizam GPS (a) sistema GIB (b)boias (c) pinger

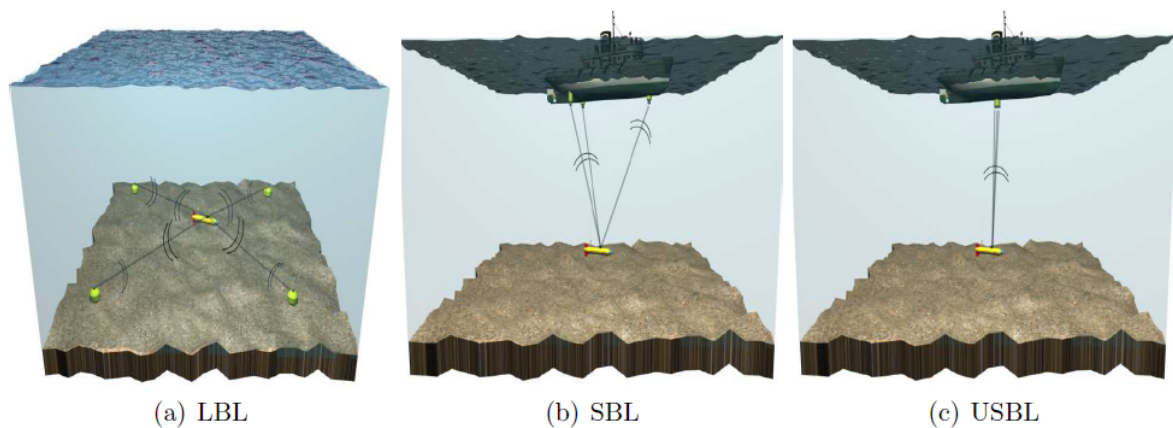


Fonte: ALCOCER; OLIVEIRA; PASCOAL,( 2006).

Já EROL; VIEIRA; GERLA, 2007 desenvolveram um método, no qual o AUV utiliza GPS e sensores submersos para estimar a sua localização. O veículo capta o sinal de GPS com a sua posição enquanto estiver emergido, então mergulha a uma profundidade fixa e realiza um caminho predefinido movendo-se através dos sensores. Durante esse período em que o AUV estiver submerso a sua posição é estimada através dos sensores (EROL; VIEIRA; GERLA, 2007). Mas essa técnica está restrita a uma trajetória e profundidade predefinidas.

Outro tipo de sensores acústicos que são utilizados para localização de AUVs, são o *Long Base Line* (LBL), *Short Base Line* (SBL) e *ultra-short base line* (USBL). Esses dispositivos estimam a posição através dos tempos de chegada das mensagens detectadas por *receivers*. O sistema LBL (Figura 4 (a)) é formado por *transponders* que são colocados no chão do oceano formando um perímetro de trabalho. Já o SBL e USBL (Figuras 4(b) e (c)) utilizam um navio que acompanha o AUV, com um *transponder* direcionado para o mesmo, e com isso consegue calcular a distância do veículo até a embarcação (VICKERY, 1998). Esse tipo de sistema foi utilizado por (SMITH; KRONEN, 1997) para determinar a posição de um AUV.

Figura 4 – Sistema acústico (a) *Long Base Line* (b) *Short Base Line* (c) *Ultra-Short base line*



Fonte: ALCO CER; OLIVEIRA; PASCOAL, (2006).

A precisão desses sensores acústicos está ligada a fatores como a escolha do lugar de fixação dos *transponders*, que podem ser fixos no solo submarino, em um navio ou em boias na superfície da água. Outro fator é o conhecimento da velocidade do som na água, devido que essa propriedade pode variar de acordo com fatores ambientais como temperatura e densidade. Caso isso não for respeitado esses dispositivos podem ficar suscetíveis a erros de *delay* (KINSEY; EUSTICE; WHITCOMB, 2006).

Também é possível estimar a localização de veículos aquáticos por meio de sensores inerciais, esse método é chamado de *dead reckoning* (DR). O DR determina a localização do veículo através de sua velocidade ou aceleração, para isso utiliza sensores como acelerômetros e giroscópios. Dessa maneira um sistema de localização baseado em sensores inerciais utiliza a integração dos valores medidos por acelerômetros e giroscópios. Contudo um dos problemas de utilizar esse tipo de abordagem é que esse sensores sofrem com erros de *drift* ao longo do tempo (PAULL et al., 2014).

Todas essas técnicas abordadas acima possuem seus prós e contras, e por isso pode-se dizer que utilizar apenas um ou outro sensor específico não garante a robustez e precisão necessária para um sistema de localização submerso. Por esse motivo utilizar a fusão de dados de vários sensores é a alternativa que melhor se encaixa para o caso de um veículo aquático. A técnica de fusão de sensores apresenta alguns benefícios como (RIGBY; PIZARRO; WILLIAMS, 2006):

- Uma melhora na precisão: A incerteza e o ruído contido nas medições podem ser reduzidos devido ao fato de que a informação é oriunda de diversas fontes.
- Robustez: A ocorrência de falhas nos sensores pode não ser tão crítico, e medições corrompidas e valores errôneos podem ser facilmente detectados.
- Aprimoramento da Resolução: sensores distintos possuem resoluções e erros temporais diferentes. Por esse motivo é possível obter uma quantidade de informação não obtível utilizando apenas um sensor.

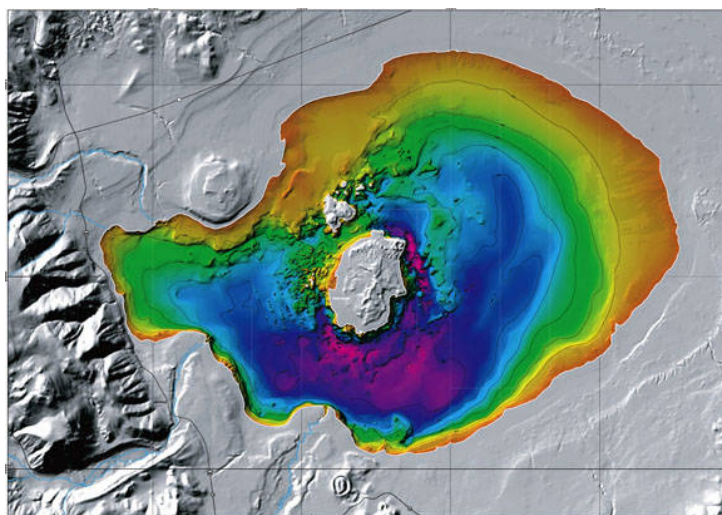
Para realizar a fusão de informação de vários sensores deve-se utilizar estimadores de estados, os quais utilizam da teoria de probabilidade para poder estimar os estados de um sistema. Os estimadores paramétricos são aqueles que suas funções de probabilidades são definidas através de parâmetros como média ( $\mu$ ) e variância ( $\sigma$ ). Dentro dessa classificação, os filtros mais conhecidos fazem parte da família gaussiana, e suas técnicas mais famosas são o Filtro de Kalman e suas variações. Já filtros não paramétricos não dependem de parâmetros, sua função de probabilidade evolui para melhor adaptar-se aos dados, um dos métodos mais conhecidos dentro dessa classificação é o Filtro de Partículas.

Um dos processos mais conhecidos para efetuar a fusão de informação de sensores é o Filtro de Kalman (FK), mas esse método serve apenas para sistemas cuja dinâmica é linear. Porém, a localização de veículos aquáticos envolve não linearidades, assim devemos utilizar o Filtro de Kalman Estendido (FKE). Por exemplo, GRENON et al., 2001 apresenta um sistema de localização, o qual utiliza uma IMU (*Inertial Measurement Unit*) e um sonar DVL (*Doppler Velocity Log*), e realiza a união dos dados através do FKE.

O Filtro de Partículas é outra técnica que tem sido utilizada para realizar a união de dados de diversos sensores. NAKATANI et al., 2009 apresenta um sistema de localização baseado no terreno (Figura 6), esse tipo de técnica utiliza mapas batimétricos (Figura 5), os quais são gráficos que indicam a profundidade de oceanos lagos e rios. Para determinar a posição do veículo em relação ao mapa é empregado o FP, uma IMU e um sonar de varredura montados no veículo. Com a fusão desses dados e com as medições dos sensores inerciais através do FP, se obtém uma estimacão dos estados precisa e robusta como é mostrado em (NAKATANI et al., 2009; ANONSEN; HALLINGSTAD, 2006; DONOVAN, 2012). Mas também existem outras alternativas como a abordada por YUN et al., 1999 que

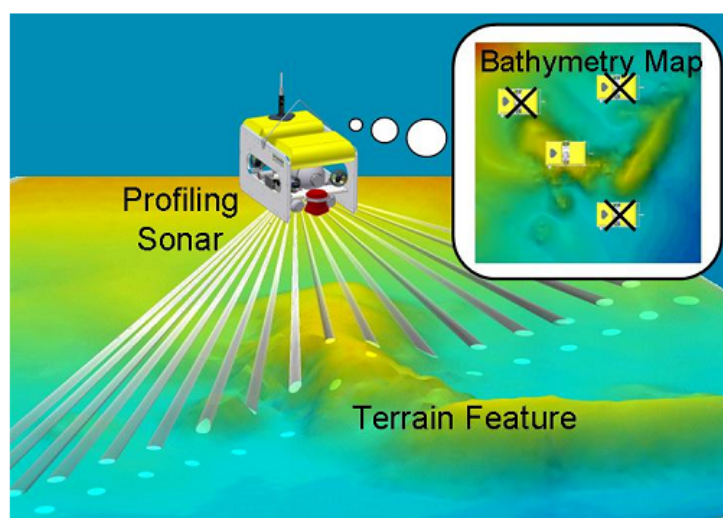
apresenta um sistema de localização que utiliza um GPS integrado com um INS (*Inertial Navigation System*), onde para realizar a fusão desses dois sensores os autores utilizam um filtro chamado *12-state filter*.

Figura 5 – Mapa batimétrico do *Mono Lake* califórnia



Fonte: RAUMANN et al., (2012).

Figura 6 – Localização baseada no terreno



Fonte: NAKATANI et al., (2009).

## 2.2 Filtro de Kalman

Certamente a técnica de estimação de estados de sistemas dinâmicos mais conhecida é o chamado Filtro de Kalman (FK). O FK separa o problema de estimação de estados em duas partes, a predição – onde o modelo matemático do sistema é utilizado – e a correção onde informações sobre os sensores são exploradas. Esta técnica de estimação pertence a família de filtros Bayesianos paramétricos, por parametrizar as incertezas do sistema e dos sensores na forma de distribuições Gaussianas, descritas por suas médias e covariâncias (THRUN; BURGARD; FOX, 2005). Para realizar a estimação dos estados o Filtro de Kalman (FK) supõem que o sistema é linear e invariante no tempo, assim considerando o seguinte sistema dinâmico discreto:

$$\begin{cases} \mathbf{x}_k = A_{k-1}\mathbf{x}_{k-1} + B_{k-1}u_k + \varepsilon_r, \\ z_k = C_k\mathbf{x}_k + \varepsilon_q, \end{cases} \quad (2.1)$$

onde,  $\mathbf{x}_k \in \mathbb{R}^n$  é o vetor de estados do sistema,  $u_k \in \mathbb{R}^m$  é o vetor de entradas e  $z_k \in \mathbb{R}^p$  é o vetor de saídas do medidas. As matrizes  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$  e  $C \in \mathbb{R}^{p \times n}$  definem o sistema. A incerteza de transição dos estados é representada por  $\varepsilon_r$  que é um sinal Gaussiano com média zero e covariância  $R_k \in \mathbb{R}^{n \times n}$ . O termo  $\varepsilon_q$  descreve o ruído inerente aos sensores utilizados e este sinal é Gaussiano com média zero e covariância  $Q_k$ . O estado atual do sistema  $\mathbf{x}_k$  é determinado pela seguinte distribuição Gaussiana:

$$\mathbf{P}(\mathbf{x}_k) \sim \mathcal{N}(\mu_k, \sigma_k)$$

de forma que  $\mu_k \in \mathbb{R}^n$  representa o valor esperado (média) de  $x_k$  e  $\sigma_k \in \mathbb{R}^{n \times n}$  sua covariância. Assim define-se as estimativas  $\hat{x}_k \triangleq \mu_k$  e  $\hat{P}_k \triangleq \sigma_k$ , conforme proposto por Kalman as estimativas *a priori* (predição) dos estados  $\bar{\mathbf{x}}_k$  e sua covariância  $\bar{P}_k$  é calculada pela seguinte expressão:

$$\begin{aligned} \mathbf{x}_k &= A_{k-1}\mathbf{x}_{k-1} + B_k u_k \\ \bar{P}_k &= A_k \bar{P}_{k-1} A_k' + R_k \end{aligned} \quad (2.2)$$

Enquanto que a estimativa final dos estados  $\hat{\mathbf{x}}_k$  e sua covariância  $\hat{P}_k$  *a posteriori* (correção) é dado por:

$$\begin{aligned} \hat{\mathbf{x}}_k &= \bar{\mathbf{x}}_k + K_k(C_k \bar{\mathbf{x}}_k - z_k^i), \\ \hat{P}_k &= (I - K_k C_k) \bar{P}_k, \end{aligned} \quad (2.3)$$

de modo que  $K_k$  representa o ganho de correção de Kalman é calculado pela seguinte expressão:

$$K_k = \bar{P}_k C_k' (C_k \bar{P}_k C_k' + Q_k)^{-1} \quad (2.4)$$



Desse modo o algoritmo do Filtro de Kalman é apresentado na Figura 7 (THRUN; BURGARD; FOX, 2005) que recebe como entrada a estimativa dos estados passada  $\hat{\mathbf{x}}_{k-1}$ , a  $\hat{P}_{k-1}$  passada, e as entradas  $u_k$  e medidas dos sensores  $z_k$  atuais. Esse algoritmo é dividido em duas etapas que são:

- **Predição:** Nas linhas 2 e 3 do algoritmo da Figura 7 é realizada a predição dos estados  $\bar{\mathbf{x}}_k$  e de sua covariância  $\bar{P}_k$ , nessa etapa o filtro utiliza o modelo do sistema.
- **Correção:** Entre as linhas 4 e 6 é realizada a correção da estimação é nessa etapa que o ganho de Kalman  $K_k$  é calculado. A medição dos sensores  $z_k$  é incorporada na estimação nesta etapa, isso é feito através do calculo do ganho  $K_k$  na linha 4. Assim nas linha 5 e 6 respectivamente são calculados a estimação final  $\hat{\mathbf{x}}_k$  e sua covariância  $\hat{P}_k$  que são armazenados para o próximo passo do filtro.

Figura 7 – Algoritmo do Filtro de Kalman.

---

```

1: procedure FK( $\hat{\mathbf{x}}_{k-1}, \hat{P}_{k-1}, u_k, z_k$ )
2:    $\bar{\mathbf{x}}_k = A_{k-1}\hat{\mathbf{x}}_{k-1} + B_k u_k$ 
3:    $\bar{P}_k = A_k \hat{P}_{k-1} A_k' + R_k$ 
4:    $K_k = \bar{P}_k C_k' (C_k \bar{P}_k C_k' + Q_k)^{-1}$ 
5:    $\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + K_k (C_k \bar{\mathbf{x}}_k - z_k^i)$ 
6:    $\hat{P}_k = (I - K_k C_k) \bar{P}_k$ 
7:   return  $\hat{\mathbf{x}}_k, \hat{P}_k$ 
8: end procedure

```

---

Fonte: THRUN; BURGARD; FOX, (2005).

## 2.3 Filtro de Kalman Estendido (FKE)

O FK pode ser aplicado apenas para estimar estados de sistemas lineares quando trabalha-se com sistemas não-lineares deve-se utilizar a sua variação que é o Filtro de Kalman Estendido ou outras abordagens completamente diferentes. Assim o EKF assume que a probabilidade de transição entre os estados e a probabilidade das medidas são governadas por funções não lineares (THRUN; BURGARD; FOX, 2005). Neste caso considera-se a seguinte representação discreta não linear:

$$\begin{cases} \mathbf{x}_k = f(u_k, \mathbf{x}_{k-1}) + \varepsilon_r, \\ z_k = h(\mathbf{x}_k) + \varepsilon_q. \end{cases} \quad (2.5)$$

A função não-linear  $f(u_k, \mathbf{x}_{k-1}) \in \mathbb{R}^n$  representa o modelo de transição dos estados do sistema e  $h(\mathbf{x}_k) \in \mathbb{R}^p$  define os sensores de saída. Os sinais  $\varepsilon_r$  e  $\varepsilon_q$  são respectivamente as incertezas do modelo e o ruído das medições, e são sinais Gaussianos com média zero e covariância respectivamente  $R_k$  e  $Q_k$ . Então, para tratar o sistema (2.5) o FKE utiliza a aproximação de Taylor de primeira ordem para linearizar o sistema, dessa forma a função não linear  $f(u_k, \mathbf{x}_{k-1})$  fica da seguinte forma :

$$f(u_k, \mathbf{x}_{k-1}) \approx f(u_k, \bar{\mathbf{x}}_{k-1}) + \underbrace{\frac{\partial f(u_k, \bar{\mathbf{x}}_{k-1})}{\partial \bar{\mathbf{x}}_{k-1}}}_{=: F} (\mathbf{x}_{k-1} - \bar{\mathbf{x}}_{k-1}) \quad (2.6)$$

Logo a função  $F \in \mathbb{R}^{n \times m}$  representa a matriz jacobiana do sistema, ou seja, a matriz linearizada pela aproximação de Taylor. Assim a predição dos estados  $\mathbf{x}_k$  e sua covariância  $\bar{P}_k$  é determinada pela seguinte expressão:

$$\begin{aligned} \bar{\mathbf{x}}_k &= f(\mathbf{x}_{k-1}, u_{k-1}) \\ \bar{P}_k &= F_k \bar{P}_{k-1} F_k' + R_k \end{aligned} \quad (2.7)$$

Para a matriz que representa a saída dos sensores  $h(\mathbf{x}_k)$  também aplica-se a expansão de Taylor como é mostrado na equação (2.8).

$$h(\mathbf{x}_k) \approx h(\bar{\mathbf{x}}_k) + \underbrace{\frac{\partial h(\bar{\mathbf{x}}_k)}{\partial \bar{\mathbf{x}}_k}}_{=: H} (\mathbf{x}_k - \bar{\mathbf{x}}_k) \quad (2.8)$$

de forma que  $H \in \mathbb{R}^p$  representa a matriz Jacobiana dos sensores. Assim a estimativa final dos estados  $\hat{\mathbf{x}}_k$  e sua covariância  $\hat{P}_k$  e calculado pela seguinte expressão:

$$\begin{aligned} \hat{\mathbf{x}}_k &= \bar{\mathbf{x}}_k + K_k (h_k^i - z_k^i) \\ \hat{P}_k &= (I - K_k H_k) \bar{P}_k \end{aligned} \quad (2.9)$$

Com o processo de linearização o FKE se diferencia do FK normal, dessa forma a sua aplicação em sistemas não-lineares é permitida. Na Figura 8 é apresentado o algoritmo do FKE que recebe como entrada os estados estimados no passo anterior  $\hat{\mathbf{x}}_{k-1}$ , a covariância  $\hat{P}_{k-1}$ , as entradas  $u_k$  e as medidas dos sensores  $z_k$ . Esse Algoritmo funciona em duas etapas que são:

- **Predição** : Nas linhas 2 e 3 é feita a predição dos estados  $\bar{\mathbf{x}}_k$  e calculada a sua covariância  $\bar{P}_k$ , essa etapa baseia-se apenas no modelo do sistema.
- **Correção** : Entre as linhas 4 e 6 é que ocorre a correção dos estados estimados é nessa etapa que o ganho de Kalman  $K_k$  é calculado. Com isso é incorporado as medições dos sensores  $z_k$  na estimação. Esse ganho corrige o erro entre a medição

e a estimação do estado na linha 5, por ultimo, na linha 6 é calculada uma nova matriz de covariância  $\hat{P}_k$ .

Figura 8 – Algoritmo do Filtro de Kalman Estendido.

---



---

```

1: procedure FKE( $\hat{\mathbf{x}}_{k-1}, \hat{P}_{k-1}, u_k, z_k$ )
2:    $\bar{\mathbf{x}}_k = f(\hat{\mathbf{x}}_{k-1}, u_{k-1})$ 
3:    $\bar{P}_k = F_k \hat{P}_{k-1} F_k' + R_k$ 
4:    $K_k = \bar{P}_k H_k' (H_k \bar{P}_k H_k' + Q_k)^{-1}$ 
5:    $\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + K_k (h_k^i - z_k^i)$ 
6:    $\hat{P}_k = (I - K_k H_k) \bar{P}_k$ 
7:   return  $\hat{\mathbf{x}}_k, \hat{P}_k$ 
8: end procedure

```

---

Fonte: THRUN; BURGARD; FOX, (2005).

## 2.4 Filtro de Partículas

Outra técnica que tem sido cada vez mais estudada para realizar a estimação de estados é o Filtro de Partículas (FP). O FP assim como o FK e FKE, realiza a estimação dos estados em dois passos: predição e correção. Na predição pode ser utilizado um determinado tipo de sensor, que no caso deste trabalho é uma IMU, e para a correção utiliza-se a medição de outro sensor distinto, um sonar, por exemplo. Dessa forma atribuindo valores iniciais para um grupo de partículas através do primeiro sensor (IMU), e corrigindo esses valores atribuindo pesos de importância para as mesmas utilizando a medição do segundo sensor (sonar) é possível obter uma estimação precisa do estado.

Assim como as outras técnicas de estimação apresentadas (FKE e FKE) o FP é uma variação do filtro de *Bayes*, é classificado como um filtro não paramétrico, isso porque não depende de parâmetros como a  $\mu$  (média) e  $\sigma$  (variância), já que a sua função de probabilidade evolui com o objetivo de melhor adaptar-se aos dados. A ideia principal do FP é representar o conhecimento a *posteriori* dos estados  $\hat{\mathbf{x}}_t$  por um grupo de amostras aleatórias denominadas “partículas”. Ao contrário dos métodos gaussianos que representam a distribuição de probabilidade através da função exponencial, o FP utiliza um grupo de amostras (partículas). Dessa maneira consegue representar essa distribuição por um espaço maior que as técnicas gaussianas (THRUN; BURGARD; FOX, 2005). As

partículas são representadas da seguinte maneira:

$$\chi_t := [\mathbf{x}_t^1, \mathbf{x}_t^2, \mathbf{x}_t^3, \dots, \mathbf{x}_t^M], \quad (2.10)$$

onde, cada partícula  $\mathbf{x}_t^M$  representa uma instancia concreta dos estado no tempo  $t$ , ou seja uma hipótese sobre o que o estado real pode ser no instante de tempo  $t$ . O número de partículas presente no grupo  $\chi_t$  é representado por  $M$ , na pratica esse valor é relativamente alto, da ordem de  $M = 1000$  ou  $M = 3000$ . Então o FP determina a estimação dos estados  $\hat{\mathbf{x}}_t$  através de um grupo de partículas  $\chi_t$ , para isso a probabilidade de transição dos estados é determinada pela regra de *Bayes* (THRUN; BURGARD; FOX, 2005):

$$\mathbf{x}_t^M \sim \mathbf{P}(\mathbf{x}_t | z_{1:t}, u_{1:t}), \quad (2.11)$$

em que,  $z_t$  representa as medidas dos sensores empregados, e  $u_t$  os sinais de controle. Com a definição da equação (2.11) pode-se dizer que, quanto mais a região do espaço de estados for povoado por partículas, maior será a chance do estado real estar nessa região. Assim á medida que  $M \rightarrow \infty$ ,  $\hat{\mathbf{x}}_t \rightarrow \mathbf{x}$  melhor será a convergência para o estado real. Como o FK e FKE, o FP constrói a sua estimação dos estados  $\hat{\mathbf{x}}_t$  recursivamente utilizando a estimação passada  $\hat{\mathbf{x}}_{t-1}$ . Na Figura 9 é apresentado um dos algoritmos mais básicos do FP, esse algoritmo recebe como entrada o grupo de partículas no tempo passado  $\chi_{t-1}$ , o valor do sinal de controle mais recente  $u_t$  e a ultima medição do sensor  $z_t$ . Seu funcionamento se dá da seguinte maneira:

- Na linha 2 é criado um grupo de partículas com valores iguais a zero, posteriormente esse grupo será transformada na estimativa do estado real  $\mathbf{x}_t$ .
- Na linha 4 é realizado o passo de predição do filtro, onde é gerada uma estimativa do estado  $\bar{\mathbf{x}}_t^M$  baseado no valor passado  $\mathbf{x}_{t-1}$  e no valor do sinal de controle  $u_t$ . Para isso é calculada a probabilidade  $\mathbf{P}(\mathbf{x}_t | u_t, \mathbf{x}_{t-1})$  para cada uma das partículas até  $M$  iterações. Desse modo no final deste passo uma grupo  $\bar{\chi}_t$  de partículas é obtido que representa uma estimativa *a priori*  $\bar{\mathbf{x}}_t$  do estado.
- Entre as linhas 5 e 11 o filtro realiza o passo de correção, e nessa etapa que a medição dos sensores é adicionada na estimativa. A medição do sensor  $z_t$  e incorporado ao grupo de partículas na linha 5, onde é calculado o fator de importância  $w_t^M$  (peso). Assim  $w_t^M$  é determinado através da probabilidade da medida do sensor  $z_t$  sobre a partícula, representado por  $w_t^M = \mathbf{P}(z_t | \mathbf{x}_t^M)$ . O grupo de partículas  $\hat{\chi}_t$ , as quais foram atribuídos os pesos, representam a estimativa do estado  $\mathbf{x}_t$  *a posteriori*.

- A grande ideia do filtro de partículas está nas linhas de 8 á 11, estas realizam o *resampling* ou *importance sampling*. Nessa parte o algoritmo extrai as partículas do grupo temporário  $\hat{\chi}_t$ , as partículas são extraídas de acordo com os pesos que lhe foram atribuídos. Quando a etapa de *resampling* é feita a distribuição de partículas é alterada, que antes era realizada através da estimação futura  $\bar{\mathbf{x}}_t$ , agora passa a ser feita pela *posteriori*  $\hat{\mathbf{x}}_t$ . Então, no final deste passo, temos um grupo de partículas  $\hat{\chi}_t$  de tamanho  $M$  apenas com as melhores partículas. Que foram selecionadas através dos pesos  $w_t^M$  atribuídos às mesmas, onde as melhores partículas são aquelas que possuem o maior valor de  $w_t^M$  que é normalizado e possui valores entre  $[0, 1]$ .

Figura 9 – Algoritmo do Filtro de Partículas.

---

```

1: procedure FP( $\chi_{t-1}, u_t, z_t$ )
2:    $\bar{\chi}_t = \chi_t = \emptyset$ 
3:   for  $m = 1$  to  $M$  do
4:     sample  $\bar{\mathbf{x}}_t^{[m]} \sim \mathbf{P}(\mathbf{x}_t | u_t, \mathbf{x}_{t-1}^{[m]})$ 
5:      $w_t^{[m]} = \mathbf{P}(z_t | \bar{\mathbf{x}}_t^{[m]})$ 
6:      $\hat{\chi}_t = \bar{\chi}_t + \langle \bar{\mathbf{x}}_t^{[m]}, w_t^{[m]} \rangle$ 
7:   end for
8:   for  $m = 1$  to  $M$  do
9:     draw  $i$  with probability  $\propto w_t^{[i]}$ 
10:    add  $\mathbf{x}_t^{[i]}$  to  $\hat{\chi}_t$ 
11:  end for
12:  return  $\hat{\chi}_t$ 
13: end procedure

```

---

Fonte: THRUN; BURGARD; FOX, (2005).

Então podemos resumir as etapas do filtro de partículas em predição e correção, iguais as usadas para o FK e FKE. Na predição o FP estima os estados *a posteriori* através da estimação passada  $\hat{\mathbf{x}}_{t-1}$  e da medições dos sinais de controles  $u_t$  através da seguinte probabilidade:

$$\hat{\mathbf{x}}_t^M \sim \mathbf{P}(\mathbf{x}_t | \hat{\mathbf{x}}_{t-1} u_t).$$

No passo de correção o FP utiliza as medições dos sensores  $z_t$  e a predição dos estados  $\bar{\mathbf{x}}_t$  para atribuir pesos de importância para cada uma das partículas através da seguinte probabilidade:

$$w_t^{[m]} \sim \mathbf{P}(z_t | \bar{\mathbf{x}}_t^M),$$

mas, a correção é realmente adicionada a estimativa final apenas quando a seleção das melhores partículas é realizada, que é feito pelo método de *Importance Resampling*. Geralmente essa seleção é realizada com uma determinada frequência, com o intuito de permitir que as partículas se propagem e adquiram a maior quantidade de informação. Uma das maneiras mais simples de implementar a técnica de *Importance Resampling* e através do algoritmo *Sampling Wheel* que é o utilizado neste trabalho para implementar o FP (THRUN; BURGARD; FOX, 2005), o qual é apresentado e discutido na Seção 4. Assim após o passo de correção a estimativa final dos estados  $\tilde{\mathbf{x}}_t$  do sistema é dado pela média de todas as partículas no instante de tempo  $t$ ,

$$\tilde{\mathbf{x}}_t = \frac{1}{M} \sum_{i=1}^M w_t^i \hat{\mathbf{x}}_t^i, \quad (2.12)$$

de forma que, a estimativa final determinada pelo FP é  $\tilde{\mathbf{x}}_t$ ,  $\hat{\mathbf{x}}_t^i$  representa o estado estimado a *posteriori* por cada partícula  $i$ ,  $w_t^i$  o seu peso de importância e  $M$  o número de partículas utilizado na estimação. O algoritmo apresentado acima representa a ideia mais básica do filtro de partículas, existem diversos estudos, os quais se dedicam apenas a otimizar o filtro. Como por exemplo em (TILTON; GHIOTTO; MEHTA, 2013) que propõem um *Feedback Particle Filter* que ao acrescentar as teorias de controle eliminou a etapa de *resampling*.

### 2.4.1 Seleção das Melhores Partículas

Um dos passos mais importantes do filtro de partículas é o método de *Importance Resampling*, e nessa etapa que é feita a seleção das melhores partículas e também a correção da estimação. Pode-se dizer que essa etapa é uma implementação probabilística da ideia de Charles Darwin sobre a sobrevivência dos mais aptos, já que após executar esse passo apenas as melhores partículas sobreviverão. Na Figura 10 é possível entender a ideia do método de *Resample*, onde mostra uma função de densidade  $f$  de uma distribuição probabilística chamada de *target distribution* (a função que se deseja determinar). Assim o objetivo é obter uma amostra de  $f$ , mas essa amostra não pode ser determinada diretamente. Em vez disso, podemos gerar partículas relacionada a uma função de densidade  $g$  (Figura 10 b), a distribuição que está relacionada a  $g$  é denotada de *proposal distribution*. A densidade  $g$  deve ser tal que  $f(\mathbf{x}) > 0$  e isso implica que  $g(\mathbf{x}) > 0$ , assim existe uma probabilidade não-nula de gerar uma partícula quando se amostra de  $g$  para qualquer estado que pode ser gerado pela amostragem de  $f$ . Contudo, o grupo de partículas resultante mostrado na parte de baixo da Figura 10 b, e distribuído de acordo com  $g$  e não por  $f$ . Em particular, para qualquer intervalo de  $A \subseteq \text{Range}(X)$  a contagem empírica de partículas

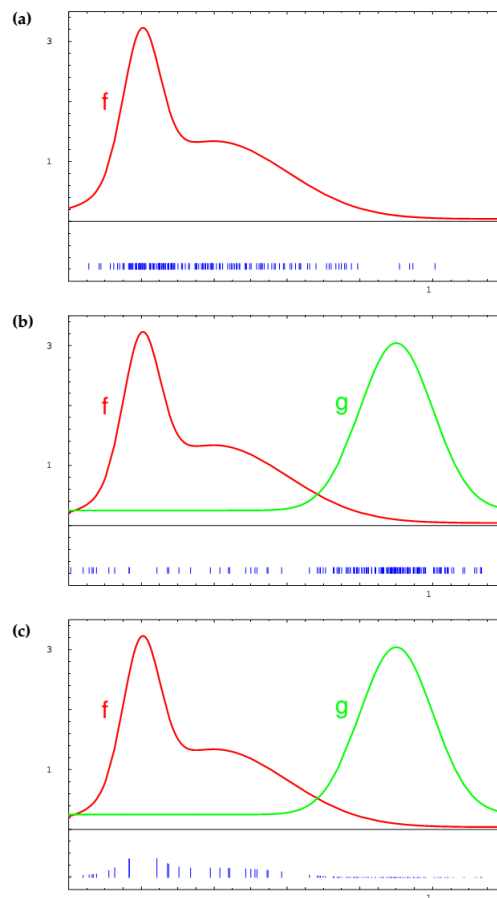
que caem em  $A$  converge para integral de  $g$  debaixo de  $A$  (THRUN; BURGARD; FOX, 2005):

$$\frac{1}{M} \sum_{m=1}^M I(\mathbf{x}^M \in A) \rightarrow \int_A g(\mathbf{x}) d\mathbf{x} \quad (2.13)$$

Mas isso gera uma diferença entre  $f$  e  $g$ , assim para compensar essa discrepância as partículas  $\mathbf{x}^M$  são ponderadas pelo seguinte quociente:

$$w^M = \frac{f(\mathbf{x}^M)}{g(\mathbf{x}^M)} \quad (2.14)$$

Figura 10 – Figura mostra a ideia por trás do método de *Resample* no FP: (a) O objetivo é aproximar a função densidade  $f$ . (b) Como não é possível gerar amostras diretamente de  $f$  utiliza-se uma função densidade diferente  $g$ , e as amostras geradas (partículas  $\mathbf{x}^M$ ) a partir de  $g$  são representadas na parte de baixo do diagrama. (c) Para obter uma amostra de  $f$  e necessário atribuir o peso de importância  $f(\mathbf{x})/g(\mathbf{x})$  para cada partícula  $\mathbf{x}^M$ . No FP  $f$  corresponde a função densidade do estado real  $\mathbf{x}$  e  $g$  ao estimado  $\hat{\mathbf{x}}$ .



Fonte: THRUN; BURGARD; FOX, (2005).

A magnitude dos pesos de importância  $w^M$  é representada pelas barras verticais da Figura 10 c, em que os pesos representam a função massa de probabilidade não-normalizada de cada partícula, assim temos que :

$$\left[ \sum_{i=1}^M w^i \right]^{-1} \sum_{i=1}^M I(\mathbf{x}^i \in A) w^i \longrightarrow \int_A f(\mathbf{x}) d\mathbf{x} \quad (2.15)$$

onde, o primeiro termo da equação atua como um normalizador para todos os pesos  $w^M$ . A equação 2.15 mostra que apesar de gerar partículas a partir da função  $g$ , as mesmas convergem para a função  $f$  quando são ponderadas adequadamente. De forma que a convergência envolve uma integração sobre todo o conjunto de  $A$ . No exemplo apresentado  $f$  é uma função contínua, um grupo de partículas representa uma distribuição discreta. Por esse motivo não existe nenhuma densidade que pode ser associada ao grupo de partículas, assim a convergência é alcançada através da função de distribuição acumulada de  $f$ . No FP a função de densidade  $f$  é o objetivo, com a assunção de que as partículas  $\chi_{t-1}$  são distribuídas de acordo com a estimativa  $\hat{\mathbf{x}}$ , a função densidade  $g$  chamada de *proposal distribution* corresponde a :

$$\mathbf{P}(\mathbf{x}_t | u_t, \mathbf{x}_{t-1}) \hat{\mathbf{x}}_{t-1}. \quad (2.16)$$

Existem diversos algoritmos que são capazes de implementar na prática a técnica de *Importance Resampling* como pode ser visto nestes trabalhos (ARULAMPALAM et al., 2002; LI; JI, 2007). Neste trabalho foi utilizado o algoritmo de *Sampling Wheel* que é considerado como um dos mais simples (THRUN; BURGARD; FOX, 2005), e seu funcionamento e aplicação é descrito na Seção 4.

## 2.5 Modelos Matemáticos

Neste seção são descritos os modelos matemáticos do veículo aquático e também dos sensores utilizados neste trabalho.

### 2.5.1 Modelo Matemático do Veículo Aquático

O veículo aquático considerado neste trabalho é o *Light Autonomous Underwater Vehicle* (LAUV) (Figura 11) é um submarino de baixo custo utilizado para pesquisas oceanográficas e do meio ambiente, seu modelo foi retirado do trabalho de SILVA et al., 2007. O LAUV possui o formato de torpedo, com 108cm de comprimento, e 15cm de diâmetro, e possui um peso de aproximadamente 18kg. Para definir o modelo do sistema dois eixos de coordenadas foram considerados: um em relação ao corpo do veículo, e outro em relação ao eixo da terra. O movimento do veículo em relação ao sistema de coordenadas



de seu corpo é descrito por seis velocidades  $\nu = [\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{p}, \mathbf{q}, \mathbf{r}]^T$ , que são respectivamente *surge*, *sway*, *heave*, *roll*, *pitch* e *yaw*. A orientação e posição do veículo em relação ao eixo de coordenadas da terra são descritos pelas seis componentes  $\eta = [x, y, z, \phi, \theta, \psi]$  (SILVA et al., 2007).

Figura 11 – Figura que mostra o veículo LAUV (parte de baixo)



Fonte: SILVA et al. (2007)

Uma vez que este trabalho tem interesse em um veículo com três graus de liberdade onde é possível representar a *pose*  $p_k = [x, y, \psi]$ , o modelo original foi simplificado para incluir apenas três  $\nu = [\mathbf{u} \ \mathbf{v} \ \mathbf{r}]^T$ . Com isso o modelo é descrito pelas velocidades em relação ao sistema de coordenadas do veículo e a *pose* em relação ao sistema de coordenadas da terra, que é,

$$p_{k+1} = p_k + T_s J(\psi_k) \nu_k, \quad (2.17)$$

onde  $T_s$  representa o tempo de amostragem e  $J(\psi_k)$  a matriz de rotação dada por:

$$J(\psi_k) = \begin{bmatrix} \cos(\psi) & -\text{sen}(\psi) & 0 \\ \text{sen}(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

As equações que descrevem o movimento do corpo são representadas por:

$$\nu_{k+1} = \nu_k + T_s M_I^{-1} (\tau_k + D(\nu_k) \nu_k) \quad (2.19)$$

em que  $M_I$  é a matriz de massa inercial do sistema e  $D(\nu_k)$  a matriz de amortecimento os valores das matrizes são encontrados em SILVA et al., 2007. As forças de propulsão do modelo referente ao eixo de coordenadas do veículo são dadas pelo vetor  $\tau_k := [F \ 0 \ \tau_\psi]^T$  na direção  $\mathbf{u}$  e entorno do torque rotacional  $\psi$  (Figura 1).

## 2.5.2 Modelo Matemático dos Sensores

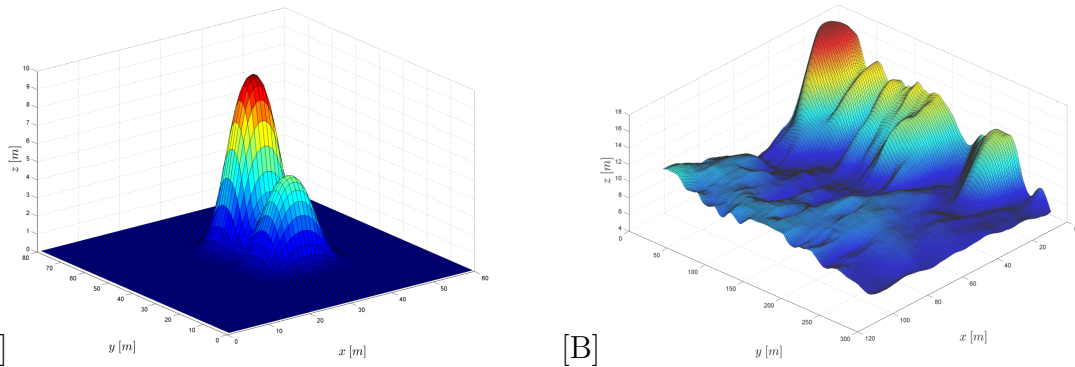
Para o desenvolvimento das técnicas de localização foram utilizados três sensores diferentes: uma IMU (*Inertial measurement unit*), sonar e sistema GIB (*GPS intelligent buoys*). A IMU empregada é composta por dois acelerômetros que medem a aceleração  $\dot{\nu} = [\dot{\mathbf{u}}, \dot{\mathbf{v}}]^T$  no sistema de coordenadas do corpo, e por um giroscópio que determina a velocidade angular  $\dot{\psi} = \mathbf{r}$ . O modelo matemático do sensor inercial é representado pelas seguintes equações:

$$\begin{aligned} a_{\mathbf{u}} &= \dot{\nu}(1) + \varepsilon_{acc} \\ a_{\mathbf{v}} &= \dot{\nu}(2) + \varepsilon_{acc} \\ g_{\psi} &= \nu(3) + \varepsilon_{gy} \end{aligned} \quad (2.20)$$

onde  $\varepsilon_{acc} \sim \mathcal{N}(\mu_{acc}, \sigma_{acc})$  e  $\varepsilon_{gy} \sim \mathcal{N}(\mu_{gy}, \sigma_{gy})$  representam o ruído branco relacionado aos acelerômetros e giroscópios respectivamente.

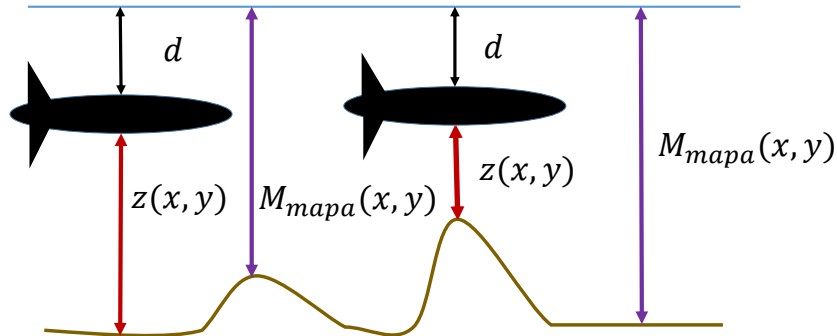
Para modelar o sonar foi considerado que a profundidade  $d$  que o veículo se encontra é conhecida (fornecida por um sensor, como um barômetro) e constante (Figura 13). Também foi utilizado um mapa batimétrico da forma de uma matriz  $M_{mapa}(m, n)$ , onde  $M_{mapa}(x, y)$  descreve a distância até o solo na coordenada  $x$  e  $y$ . Foram utilizados dois tipos de mapas um sintético criado para testes (Figura 12 A) e um real (Figura 12 B) obtido em RAUMANN et al. 2012 .

Figura 12 – A) Mapa batimétrico sintético. B) Mapa batimétrico real.



Fonte: A) O autor, (2015). B) RAUMANN et al. (2012)

Figura 13 – Representação da modelagem do sonar, a linha marrom representa o solo submarino e a linha azul a superfície da água. A seta vermelha representa a distancia medida pelo sonar  $z(x, y)$ , a seta preta mostra a profundida  $d$  do veículo, e a seta roxa descreve o valor de profundidade obtido pelo mapa  $M_{mapa}(x, y)$



Fonte: O autor, (2015)

Dessa forma o modelo matemático do sonar que determina a medida de profundidade  $z$  associada com a posição  $x$  e  $y$  do veículo é representado pela seguinte equação:

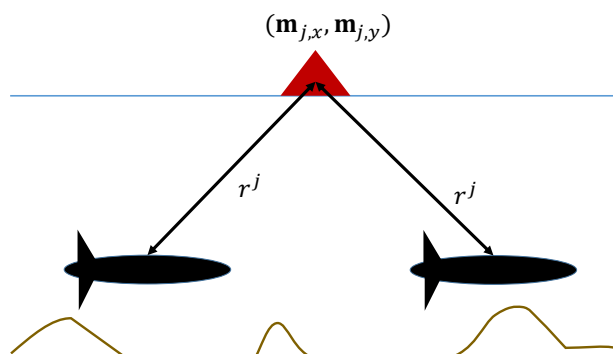
$$z(x, y) = M_{mapa}(x, y) - d + \varepsilon_z \quad (2.21)$$

de forma que  $\varepsilon_z \sim \mathcal{N}(\mu_z, \sigma_z)$  representa o ruído do sonar. Outro sensor empregado foi o sistema *GIB* que foi considerado como sendo uma boia localizada na superfície da água, onde sua posição  $(\mathbf{m}_x, \mathbf{m}_y)$  é fixa e conhecida através de um GPS (Figura 14). Esse sensor determina a distancia  $r_t$  do veículo em relação a sua posição. Essa medição é descrita pela equação 2.22.

$$r^j = \sqrt{(\mathbf{m}_{j,x} - x)^2 + (\mathbf{m}_{j,y} - y)^2 + d^2} + \varepsilon_r \quad (2.22)$$

tal que  $r^j$  representa a distância da boia em relação ao veículo e  $\varepsilon_r \sim \mathcal{N}(\mu_r, \sigma_r)$  o ruído,  $j$  e o índice da boia que é considerada. A posição da boia é dada por  $\mathbf{m}_{j,x}$  e  $\mathbf{m}_{j,y}$  e a do veículo por  $x, y$ .

Figura 14 – Representação do sistema GIB modelado, onde o triângulo vermelho representa a boia (GIB), a seta da cor preta a distancia  $r^j$  medida, a linha azul representa a superfície da água e a linha marrom o solo aquático



Fonte: o autor, (2015)

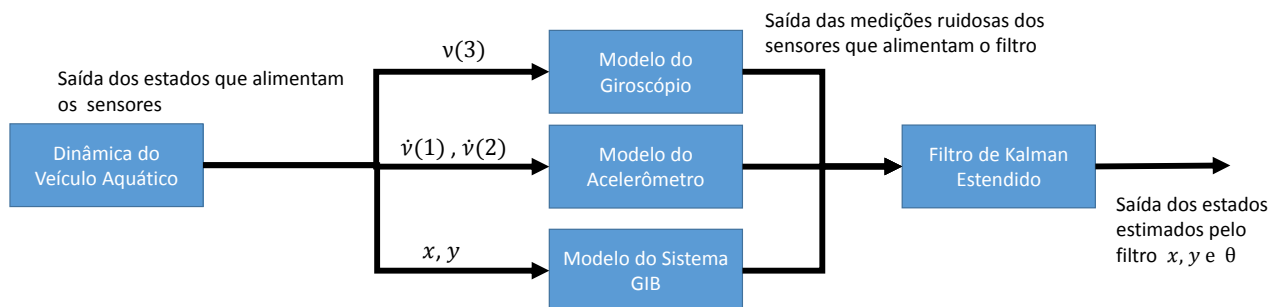
## 3 Localização Baseada no Filtro de Kalman Estendido

Neste capítulo é descrito como foi desenvolvido o Filtro de Kalman Estendido (FKE) para solucionar o problema de localizar um veículo aquático. No final deste capítulo são descritos os resultados obtidos utilizando o FKE, através de simulações feitas com o software *MATLAB*.

### 3.1 Implementação do Filtro de Kalman Estendido

O Filtro de Kalman Estendido funciona em dois passos, um em que ocorre a predição dos estados do sistema e no outro em que é feita a correção, para isso é necessário dois modelos. Geralmente utiliza-se o modelo dinâmico do sistema para realizar o passo de correção, mas neste trabalho optou-se em desacoplar totalmente o modelo do veículo do modelo do filtro (Figura 15). Sendo assim, para desenvolver o passo de predição utiliza-se as medições provenientes dos acelerômetros e giroscópios, já para a correção dos estados estimados é utilizado o sistema GIB (*GPS Intelligent buoys*). Nas duas próximas seções será explicado detalhadamente como foi realizado a implementação do filtro.

Figura 15 – Diagrama de blocos que representa a estrutura do FKE desenvolvido, onde  $\nu(3)$ ,  $\dot{\nu}(1)$  e  $\dot{\nu}(2)$  são respectivamente a velocidade angular e as acelerações lineares nos eixos  $x$  e  $y$  do modelo do veículo.



Fonte: o autor, (2016)

### 3.1.1 Predição

Para realizar o passo de predição do filtro o modelo  $\dot{\mathbf{x}} = f(\mathbf{x}, u)$  foi utilizado. Assim, estados do sistema são  $\mathbf{x} = [x \ \dot{x} \ y \ \dot{y} \ \psi]^T$  e as entradas do modelo são as medições da *IMU* e são representadas por  $u = [a_{\mathbf{u}} \ a_{\mathbf{v}} \ g_{\psi}]^T$ . Dessa forma a função  $f(\mathbf{x}, u)$  é descrita da seguinte maneira:

$$f(\mathbf{x}, u) = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{\psi} \end{bmatrix}, \quad (3.1)$$

em que  $\ddot{x}$ ,  $\ddot{y}$  e  $\dot{\psi}$  são descritos pelas seguintes equações:

$$\begin{aligned} \ddot{x} &= \cos(\hat{\psi})a_{\mathbf{u}} - \sin(\hat{\psi})a_{\mathbf{v}} \\ \ddot{y} &= \sin(\hat{\psi})a_{\mathbf{u}} + \cos(\hat{\psi})a_{\mathbf{v}} \\ \dot{\psi} &= g_{\psi} \end{aligned} \quad (3.2)$$

onde foram obtidos através da aplicação da equação (2.20) na matriz de rotação da expressão (2.18). As medidas de aceleração oriundas dos acelerômetros são  $a_{\mathbf{u}}$ ,  $a_{\mathbf{v}}$ , e  $g_{\psi}$  é a velocidade angular medida pelo giroscópio. Com esse modelo de predição a implementação do filtro está totalmente desacoplada do modelo do sistema. Assim, aplica-se a aproximação de Taylor no modelo (3.1) e obtém-se a Jacobiana  $F_k$  do modelo de predição não-linear:

$$F_k = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \varrho_1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \varrho_2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \varrho := \begin{bmatrix} -\sin(\hat{\psi})a_{\mathbf{u}} - \cos(\hat{\psi})a_{\mathbf{v}} \\ \cos(\hat{\psi})a_{\mathbf{u}} - \sin(\hat{\psi})a_{\mathbf{v}} \end{bmatrix} \quad (3.3)$$

Dessa forma, utilizando o modelo de predição  $f(x_k, u_k)$  e sua Jacobina  $F_k$  determina-se a predição dos estados  $\bar{\mathbf{x}}$  e sua covariância  $\bar{P}_k$ .

$$\begin{aligned} \bar{\mathbf{x}}_k &= f(x_k, u_k)T_s + \bar{\mathbf{x}}_{k-1} \\ \bar{P}_k &= F_k P_{k-1} F_k' + R_k \end{aligned} \quad (3.4)$$

em que,  $R_k$  é um parâmetro ajustável e representa a matriz de incertezas do modelo de predição e  $T_s$  é o período de amostragem.

### 3.1.2 Correção

Para realizar o passo de correção foi utilizado o sensor GIB (boia), assim o modelo de correção  $\hat{z}_k^i$  empregado no FKE foi o seguinte:

$$h_k^i = \left[ \sqrt{\tau} \right] \quad (3.5)$$

onde,  $i = [1, 2, \dots, i_{max}]$  representa o índice da boia que está sendo considerada, e  $\tau$  descrito por:

$$\tau_i = (\mathbf{m}_{i,x} - \bar{x}_k)^2 + (\mathbf{m}_{i,y} - \bar{y}_k)^2$$

em que,  $\mathbf{m}_{i,x}$  e  $\mathbf{m}_{i,y}$  são respectivamente as posições  $x$  e  $y$  da boia, e  $\bar{x}_k$  e  $\bar{y}_k$  são os estados preditos no passo anterior. Para encontrar a Jacobiana  $H_k^i$  referente ao modelo de correção foi utilizado a aproximação de Taylor, e dessa forma obteve-se a seguinte matriz:

$$H_k^i = \begin{bmatrix} \frac{-(m_{i,x} - \bar{x}_k)}{\sqrt{(\tau_i)}} & 0 & \frac{-(m_{i,y} - \bar{y}_k)}{\sqrt{(\tau_i)}} & 0 & 0 \end{bmatrix} \quad (3.6)$$

Dessa forma, utilizando a expressão (3.5) e a Jacobiana definida em (3.6) implementou-se o passo de correção do FKE com as seguintes equações:

$$\begin{aligned} K_k &= \bar{P}_k H_k' (H_k \bar{P}_k H_k' + Q_k)^{-1} \\ \hat{\mathbf{x}}_k &= \bar{\mathbf{x}}_k + K_k (h_k^i - z_k^i) \\ \hat{P}_k &= (I - K_k H_k) \bar{P}_k \end{aligned} \quad (3.7)$$

onde,  $K_k$  é o ganho do filtro,  $Q_k$  é um parâmetro ajustável e representa a matriz de incertezas dos sensores utilizados na correção,  $I$  é matriz identidade, e  $z_k^i$  é a medição real do sensor GIB definido na equação (2.22). Os estados estimado pelos filtro no passo de correção são definidos por  $\hat{x}_k$  e a sua matriz de covariância  $\hat{P}_k$ .

Com isso implementou-se o algoritmo apresentado na Figura 16 referente a localização utilizando o FKE. Esse código recebe como entrada as posições  $m_x$  e  $m_y$  de todas as boias utilizadas, as matrizes de incertezas  $Q_k$  e  $R_k$ , e a medição real do sistema GIB  $Z_k^i$ , e retorna como saída os estados estimados  $\hat{\mathbf{x}}_k$ . Na linha 2 é calculada a Jacobiana  $F_k$  do modelo, depois entre as linhas 3 e 4 é calculado a predição  $\bar{\mathbf{x}}_k$  e sua covariância  $\bar{P}_k$ . Depois entre as linhas 5 até 9 é calculado o modelo de correção  $\hat{h}_k^i$  e sua Jacobiana  $H_k^i$  para cada boia de  $i = 1$  até  $i_{max}$  que é o número máximo de boias consideradas. Assim na linha 10 é calculado o ganho  $K_k$  do filtro, e nas últimas linhas 11 até 12 é feito o calculo da estimação final dos estados  $\mathbf{x}_k$  e sua covariância  $P_k$ .

Figura 16 – Algoritmo de localização utilizando o Filtro de Kalman Estendido.

---

```

1: procedure FKE( $\mathbf{m}_x, \mathbf{m}_y, Z_k$ )
2:    $f(\mathbf{x}_{k-1}, u_{k-1}) = [\dot{x} \ \ddot{x} \ \dot{y} \ \ddot{y} \ \dot{\psi}]^T$ 
3:    $F_k = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \varrho_1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \varrho_2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ 
4:    $\bar{\mathbf{x}}_k = f(x_{k-1}, u_{k-1})\mathbf{T}s + \bar{\mathbf{x}}_{k-1}$ 
5:    $\bar{P}_k = F_k \bar{P}_{k-1} F_k' + R_k$ 
6:   for  $i = 1$  to  $i_{max}$  do
7:      $\tau^i = (\mathbf{m}_x^i - \bar{x}_k)^2 + (\mathbf{m}_y^i - \bar{y}_k)^2$ 
8:      $h_k^i = \sqrt{\tau^i}$ 
9:      $H_k^i = \begin{bmatrix} \frac{-(\mathbf{m}_x^i - \bar{x})}{\sqrt{(\tau^i)}} & 0 & \frac{-(\mathbf{m}_y^i - \bar{y})}{\sqrt{(\tau^i)}} & 0 & 0 \end{bmatrix}$ 
10:  end for
11:   $K_k = \bar{P}_k H_k' (H_k \bar{P}_k H_k' + Q_k)^{-1}$ 
12:   $\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + K_k (h_k^i - Z_k^i)$ 
13:   $\hat{P}_k = (I - K_k H_k) \bar{P}_k$ 
14:  return  $\hat{\mathbf{x}}_k$ 
15: end procedure

```

---

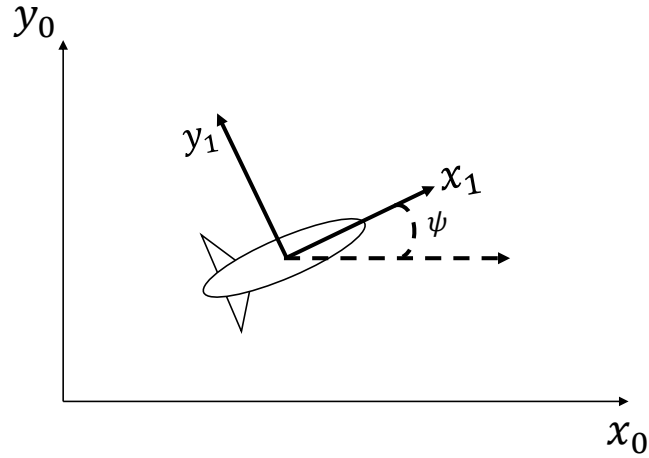
Fonte: o autor, (2016).

## 3.2 Correção utilizando um Sensor Virtual

Com o objetivo de melhorar o desempenho do método de localização utilizando o FKE descrito acima, foi modelado um sensor virtual que considera a taxa de movimento do sistema, o qual foi adicionada ao modelo de correção do filtro. Assim considerando a Figura 17 existem dois eixos de coordenadas, o global  $x_0$  e  $y_0$  e o do corpo do sistema  $x_1$ , e  $y_1$ .



Figura 17 – Representação do veículo nos dois eixos de coordenadas.



Fonte: o autor, (2016)

Sendo assim pode-se descrever a taxa de movimento do sistema no eixo global  $\nabla v_0$  pela seguinte expressão:

$$\nabla v_0 = R(\hat{\psi})V_1 \quad (3.8)$$

onde,  $R(\hat{\psi}) = \begin{bmatrix} \cos(\hat{\psi}) & -\sin(\hat{\psi}) \\ \sin(\hat{\psi}) & \cos(\hat{\psi}) \end{bmatrix}$  é a matriz de rotação, e  $V_1 = [1 \ 0]^T$  é o vetor de velocidades no eixo de coordenadas do sistema. Desse modo a taxa de movimento é definida por:

$$\nabla v_0 = \begin{bmatrix} \cos(\hat{\psi}) \\ \sin(\hat{\psi}) \end{bmatrix} \quad (3.9)$$

onde, a medição do sensor virtual é determinada pela seguinte expressão:

$$Z_V = \nabla v_0 - \begin{bmatrix} V_x \\ V_y \end{bmatrix} = 0 \quad (3.10)$$

tal que,  $V_x$ ,  $V_y$  são definidos da seguinte maneira:

$$V_x = \frac{\hat{x}_k}{|\hat{x}_k||\hat{y}_k|}, \text{ se } \hat{x}_k > 0$$

$$V_y = \frac{\hat{y}_k}{|\hat{x}_k||\hat{y}_k|}, \text{ se } \hat{y}_k > 0 \quad (3.11)$$

onde,  $\hat{x}_k$  e  $\hat{y}_k$  são as velocidades estimadas pelo FKE. Com o modelo do sensor virtual definido pela expressão (3.10), adiciona-se os dois estados ao modelo do sensor real  $Z_k^i$  referentes ao sensor virtual. Assim o novo modelo do sensor real é definido da seguinte forma:

$$Z_k^i = \begin{bmatrix} \sqrt{(\mathbf{m}_x^i - x)^2 + (\mathbf{m}_y^i - y)^2} + \varepsilon_r \\ 0 \\ 0 \end{bmatrix} \quad (3.12)$$

A medição do sensor virtual foi adicionada no modelo de correção  $\hat{z}_k^i$  obtendo-se assim a seguinte matriz:

$$h_k^i = \begin{bmatrix} \sqrt{\tau_k^i} \\ V_x - \cos(\bar{\psi}_k) \\ V_y - \cos(\bar{\psi}_k) \end{bmatrix} \quad (3.13)$$

Assim através da expansão de Taylor obteve-se a nova Jacobiana  $H_k^i$  definida na equação 3.14 referente ao novo modelo de correção.

$$H_k^i = \begin{bmatrix} \frac{-(\mathbf{m}_x - \bar{x})}{\sqrt{(\tau_k^i)}} & 0 & \frac{-(\mathbf{m}_y - \bar{y})}{\sqrt{(\tau_k^i)}} & 0 & 0 \\ 0 & 1 & 0 & 0 & \sin(\bar{\psi}_k) \\ 0 & 0 & 0 & 1 & -\cos(\bar{\psi}_k) \end{bmatrix} \quad (3.14)$$

O algoritmo apresentado na Figura 18 é referente ao método de localização utilizando o FKE com o sensor virtual. Na linha 1 é calculado a Jacobiana do modelo, depois nas linhas 3 e 4 é realizado a predição dos estados do sistema. Já entre as linhas 5 até 14 existe uma proteção para que o cálculo da norma não cause singularidades. O cálculo do modelo de correção é feito nas linhas 15 até 19 e, por último, nas linhas 20 até 22 é calculado o passo de correção do filtro.

Figura 18 – Algoritmo de localização utilizando o Filtro de Kalman Estendido com o sensor virtual.

---



---

```

1: procedure FKE( $\mathbf{m}_x, \mathbf{m}_y, Z_k$ )

2:    $F_k = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \varrho_1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \varrho_2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ 

3:    $\bar{\mathbf{x}}_k = f(\mathbf{x}_{k-1}, u_{k-1}) \mathbb{T}_S + \bar{\mathbf{x}}_{k-1}$ 
4:    $\bar{P}_k = F_k P_{k-1} F_k' + R_k$ 

5:   if  $|\dot{x}_k| > 0$  then
6:      $V_x = \frac{\dot{x}_k}{|\dot{x}_k| |\dot{y}_k|}$ 
7:   else
8:      $V_x = 0$ 
9:   end if

10:  if  $|\dot{y}_k| > 0$  then
11:     $V_y = \frac{\dot{y}_k}{|\dot{x}_k| |\dot{y}_k|}$ 
12:  else
13:     $V_y = 0$ 
14:  end if

15:  for  $i = 1$  to  $i_{max}$  do
16:     $\tau^i = (\mathbf{m}_x^i - \bar{x}_k)^2 + (\mathbf{m}_y^i - \bar{y}_k)^2$ 
17:     $h_k^i = \begin{bmatrix} \sqrt{\tau_k^i} \\ V_x - \cos(\bar{\psi}_k) \\ V_y - \cos(\bar{\psi}_k) \end{bmatrix}$ 
18:     $H_k^i = \begin{bmatrix} \frac{-(\mathbf{m}_x - \bar{x})}{\sqrt{(\tau_k^i)}} & 0 & \frac{-(\mathbf{m}_y - \bar{y})}{\sqrt{(\tau_k^i)}} & 0 & 0 \\ 0 & 1 & 0 & 0 & \sin(\bar{\psi}_k) \\ 0 & 0 & 0 & 1 & -\cos(\bar{\psi}_k) \end{bmatrix}$ 
19:  end for

20:   $K_k = \bar{P}_k H_k' (H_k \bar{P}_k H_k' + Q_k)^{-1}$ 
21:   $\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + K_k (h_k^i - Z_k^i)$ 
22:   $\hat{P}_k = (I - K_k H_k) \bar{P}_k$ 

23:  return  $\mathbf{x}_k$ 
24: end procedure

```

---

### 3.3 Resultados

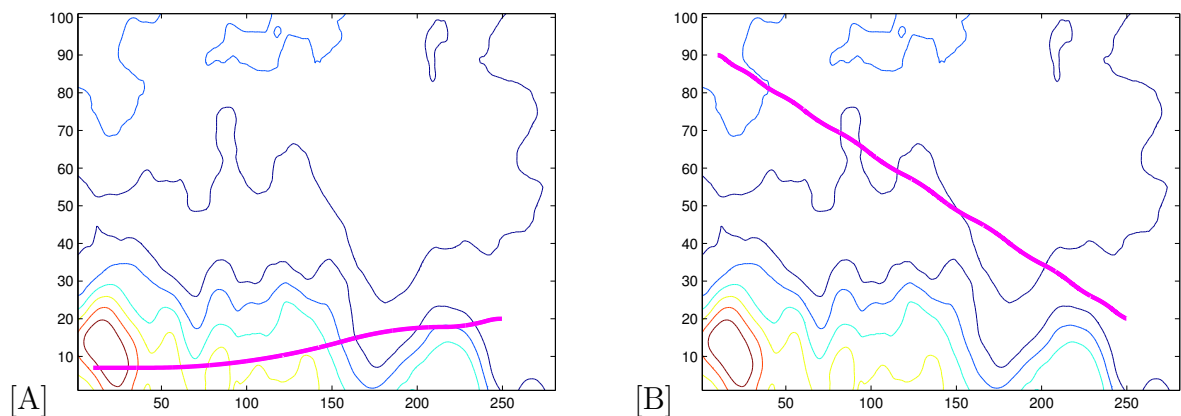
Nesta seção são apresentados os resultados obtidos utilizando a técnica de localização baseado no Filtro de Kalman Estendido, os quais foram adquiridos através de simulações realizadas no *software* MATLAB. Para determinar o número mínimo de boias necessárias para obter uma estimacão com bom desempenho, foram realizadas simulações utilizando apenas uma boia, duas, e três. Foi assumido que a posição inicial do veículo é conhecida através de um *GPS*, sendo assim as condições iniciais do filtro foram configuradas para os mesmos valores do veículo. Foram utilizadas duas trajetórias que estão representadas na Figura 19, a primeira (Figura 19 A ) com posição inicial em (10, 7) e posição final (250, 20), e a segunda (Figura 19 B ) com posição inicial em (10, 90) e final em (250, 20). Os valores utilizados para a matriz de incertezas do modelo do FKE foram  $R_k = \text{diag} \{[10^{-3} \ 10^{-3} \ 10^{-3} \ 10^{-3} \ 10^{-3}]\}$ . Para simular o ruído dos sensores foram utilizados os valores de média  $\mu$  e covariância  $\sigma$  apresentados na tabela 1.

Tabela 1 – Tabela com os valores de media ( $\mu$ ) e covariância ( $\sigma$ ) utilizados nos ruídos dos sensores

Ruído	Valores ( $\mathcal{N}(\mu, \sigma)$ )
$\varepsilon_{acc}$ (acelerômetro)	$\mathcal{N}(0.01, 0.001)$
$\varepsilon_{gy}$ (giroscópio)	$\mathcal{N}(0.01, 0.1)$
$\varepsilon_r$ (boia)	$\mathcal{N}(0.01, 0.1)$

Fonte: o autor (2016)

Figura 19 – Trajetórias utilizadas para obter os resultados utilizando a técnica de localização baseada no FKE. A) primeira trajetória. B) Segunda trajetória

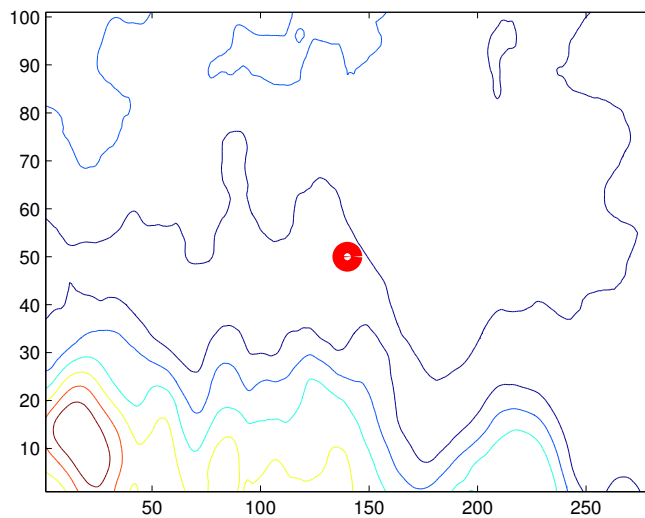


Fonte: o autor, (2016)

### 3.3.1 Resultados obtidos com uma Boia

Para o método de localização baseado no FKE com apenas uma boia foi utilizado uma matriz de incerteza dos sensores com valores de  $Q_k = \text{diag}\{[10^{-3}]\}$ . A única boia utilizada foi disposta no centro do mapa na posição de (140, 50) e está representada na Figura 20.

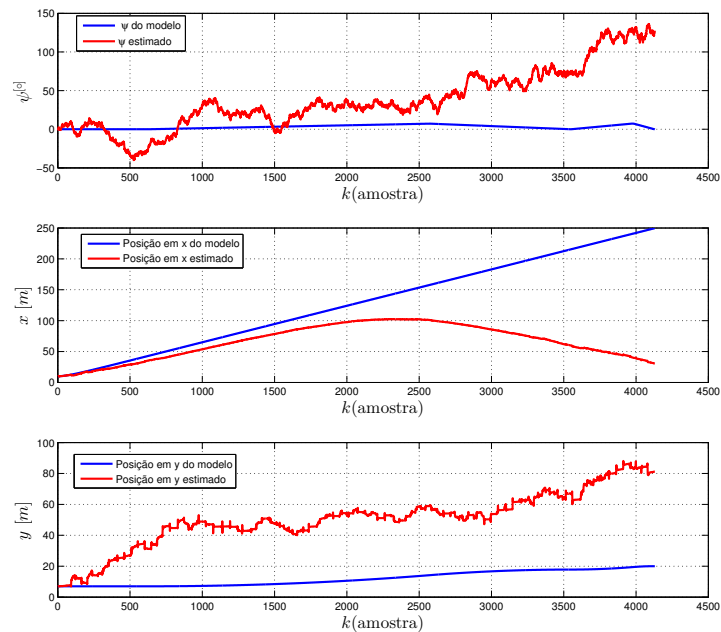
Figura 20 – Localização da boia utilizada para o método de localização baseado no FKE com uma boia.



Fonte: o autor, (2016)

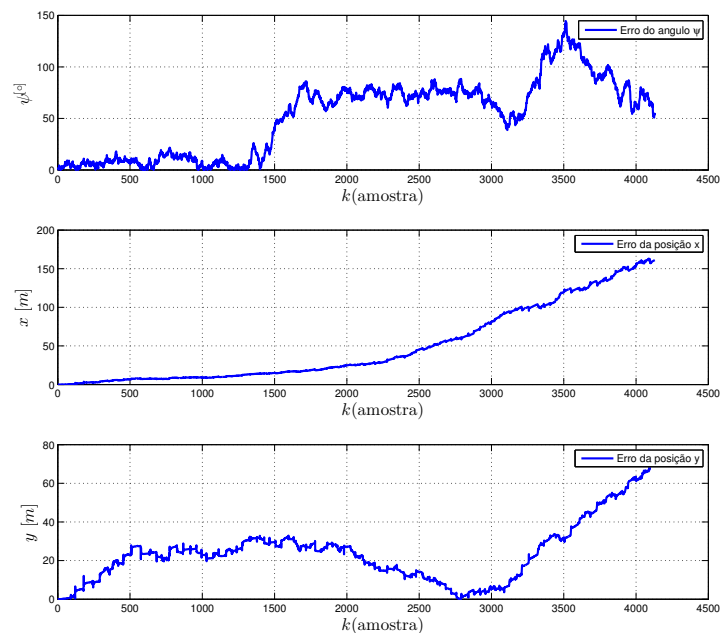
A estimação da pose  $p_k = [x \ y \ \psi]$  do veículo utilizando a técnica de localização baseada no FKE para a primeira trajetória é apresentado na Figura 21. Pode-se notar que o filtro não foi capaz de estimar nenhum dos estados utilizando apenas uma boia, isso é confirmado pela Figura 22 que mostra o erro de estimação obtido nesta abordagem.

Figura 21 – Resultado do FKE para a primeira trajetória utilizando uma boia



Fonte: o autor, (2016)

Figura 22 – Erro da estimação do filtro para a primeira trajetória

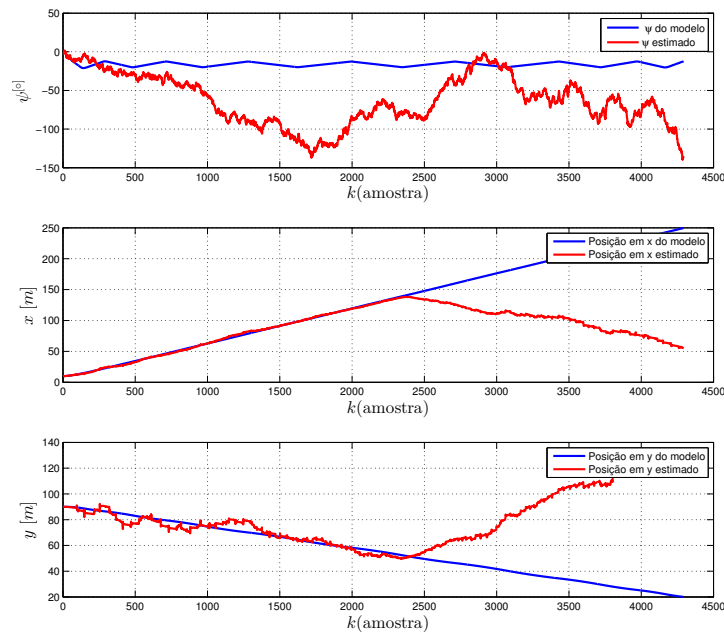


Fonte: o autor, (2016)

Para a segunda trajetória os resultados obtidos na estimação são descritos na Figura 23. Neste caso o filtro consegue estimar a posição  $x$  do veículo até a amostra  $k = 2500$ , depois desse instante a estimação perde-se. O mesmo ocorre na estimação da

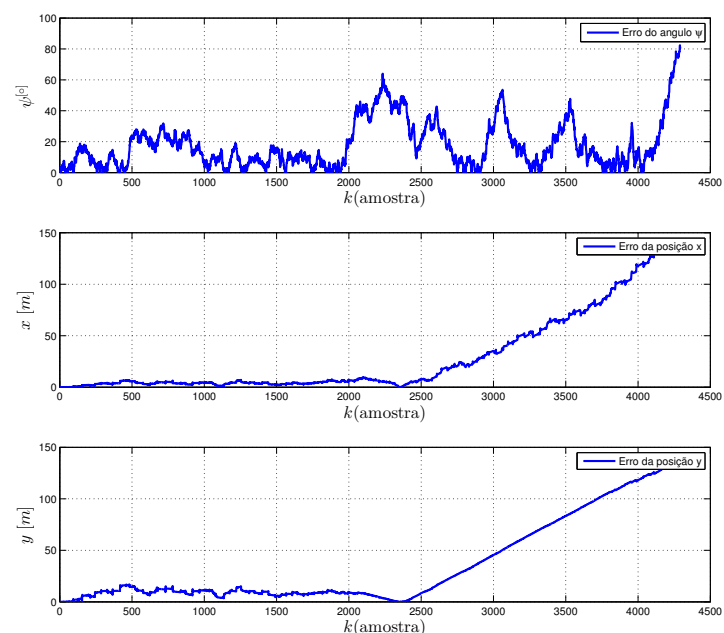
posição  $y$  que entre o instante  $k = 1000$  e  $3000$  o seu erro chega a zero como é descrito na Figura 24, mas nos momentos iniciais e final o erro de estimação é muito alto. A estimação do ângulo  $\psi$  não foi possível como é visto no gráfico do erro de estimação apresentado na Figura 24.

Figura 23 – Resultado do FKE para a segunda trajetória utilizando uma boia



Fonte: o autor, (2016)

Figura 24 – Erro da estimação do filtro para a segunda trajetória

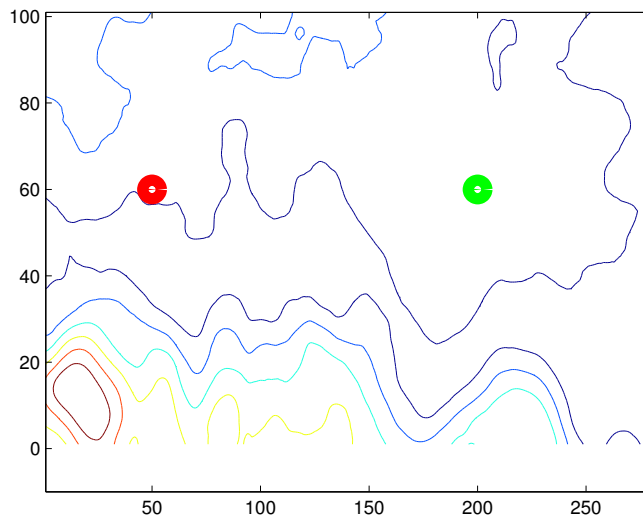


Fonte: o autor, (2016)

### 3.3.2 Resultados obtidos com duas Boia

As duas boias utilizadas foram posicionadas em (50, 60) (primeira boia cor vermelha), e (200, 60) (segunda boia cor verde) e são representadas no mapa na Figura 25. Os valores utilizados para matriz de incertezas dos sensores foi de  $Q_k = \text{diag}\{[10^{-3} \ 10^{-3}]\}$

Figura 25 – Mapa com as posições das duas boias utilizadas

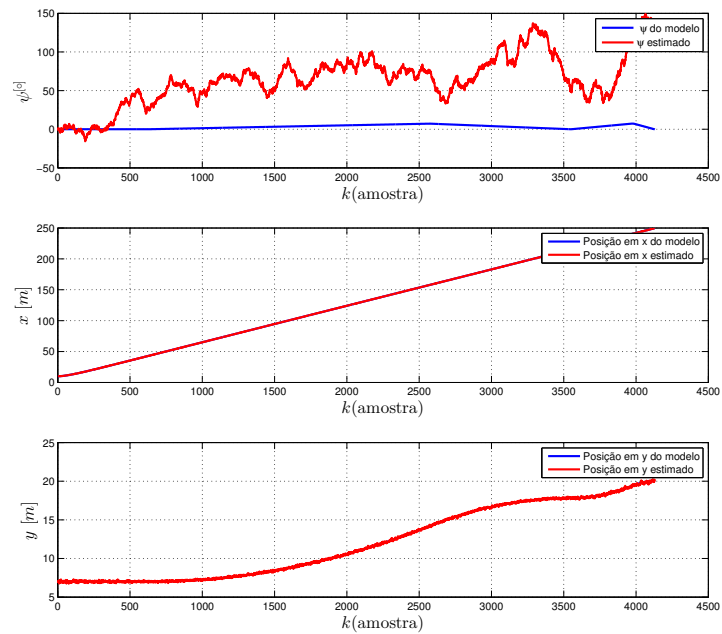


Fonte: o autor, (2016)

Os resultados de estimação obtidos para a primeira trajetória podem ser vistos na Figura 26, onde nota-se que o filtro foi capaz de determinar as posições  $x$ , e  $y$ . O erro obtido para estimação das posições  $x$ , e  $y$  foi pequeno em torno de 0.2 m como é apresentado na Figura 27. Com essa abordagem a estimação do ângulo  $\psi$  apresentou um baixo desempenho.

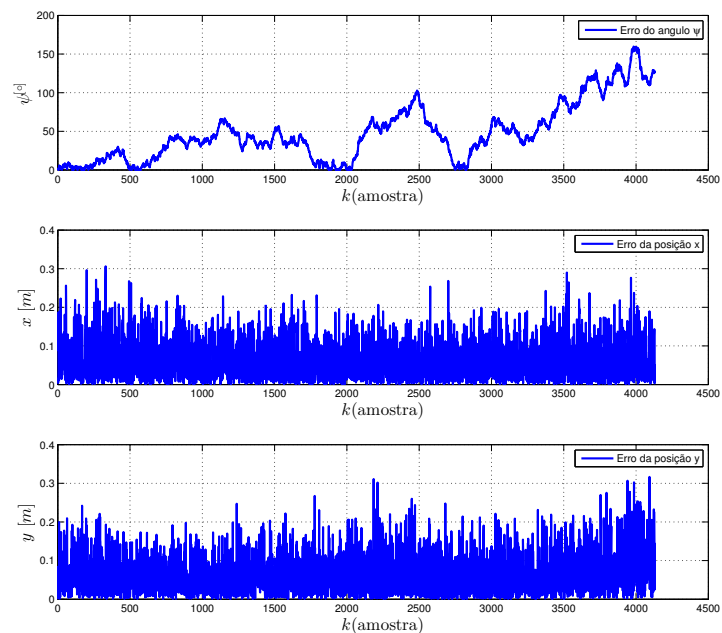


Figura 26 – Resultado do FKE para a primeira trajetória utilizando duas boias



Fonte: o autor, (2016)

Figura 27 – Erro da estimação do FKE para a primeira trajetória utilizando duas boias

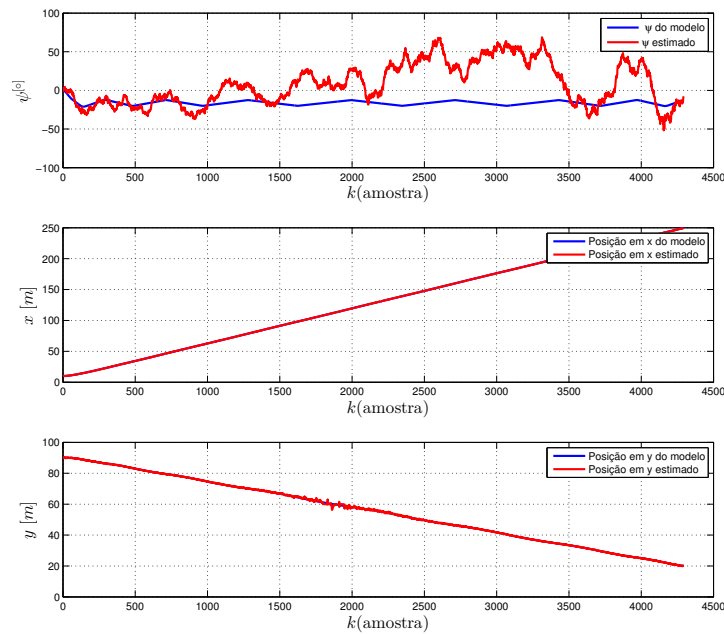


Fonte: o autor, (2016)

Para segunda trajetória os resultados da estimação para da pose  $p_k = [x \ y \ \psi]$  são apresentados na Figura 28. Foi possível obter uma estimação da posição  $x$  do veículo com um erro pequeno em torno de 0.2 m como é visto na Figura 29. Já na estimação da

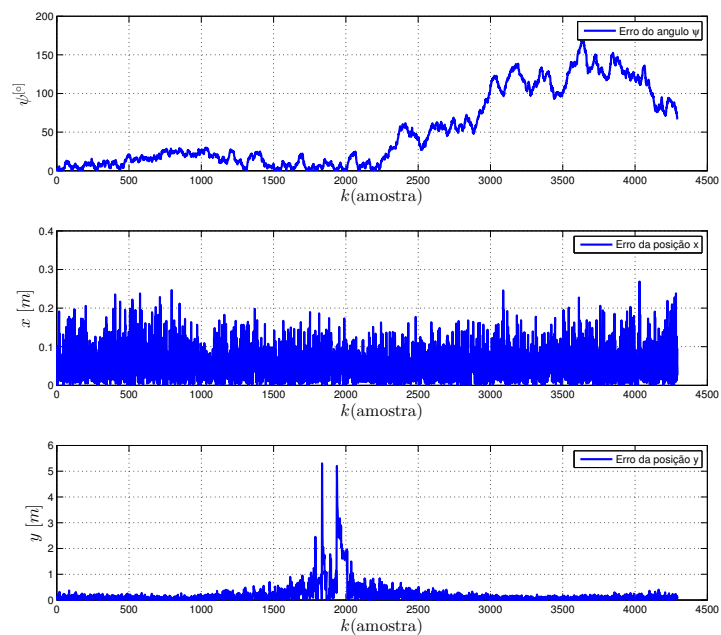
posição  $y$  ocorre um aumento no erro entre os instantes  $k = 1700$  e  $k = 2300$  como é visto no gráfico do erro da Figura 29.

Figura 28 – Resultado do FKE para a segunda trajetória utilizando duas boia



Fonte: o autor, (2016)

Figura 29 – Erro da estimação do FKE para a segunda trajetória utilizando duas boias

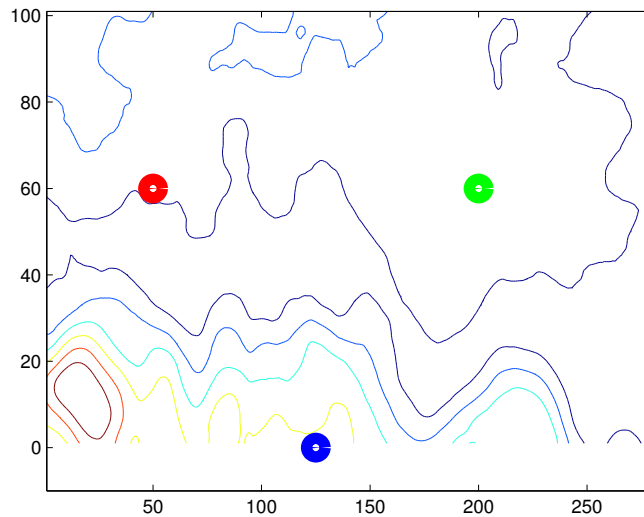


Fonte: o autor, (2016)

### 3.3.3 Resultados obtidos com três Boia

Para o método de localização baseado no FKE com três boias foi utilizado o valor de  $Q_k = \{[10^{-3} \ 10^{-3} \ 10^{-3}]\}$ . A primeira boia de cor vermelha foi situada na posição (50, 60), a segunda de cor azul foi posicionada em (125, 0), e nas coordenadas (200, 60) foi alocada a terceira boia de cor verde, as quais estão representadas no mapa da Figura 30.

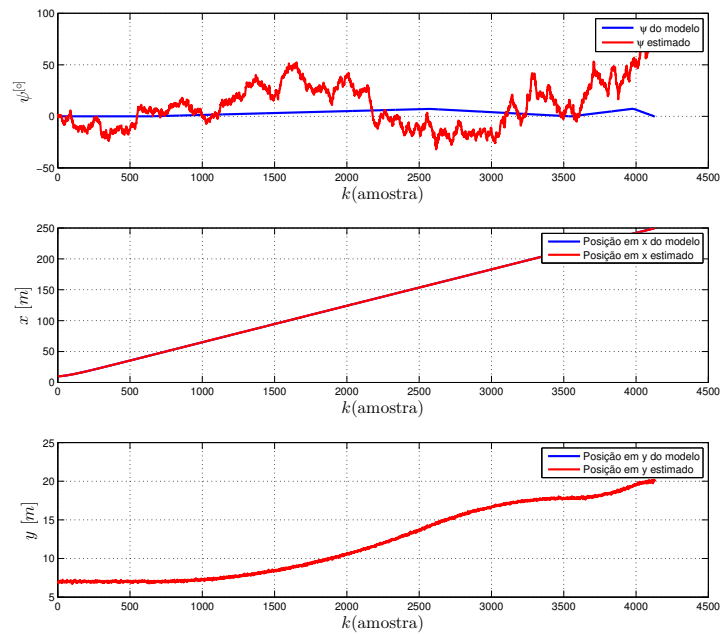
Figura 30 – Mapa com as posições das boias



Fonte: o autor, (2016)

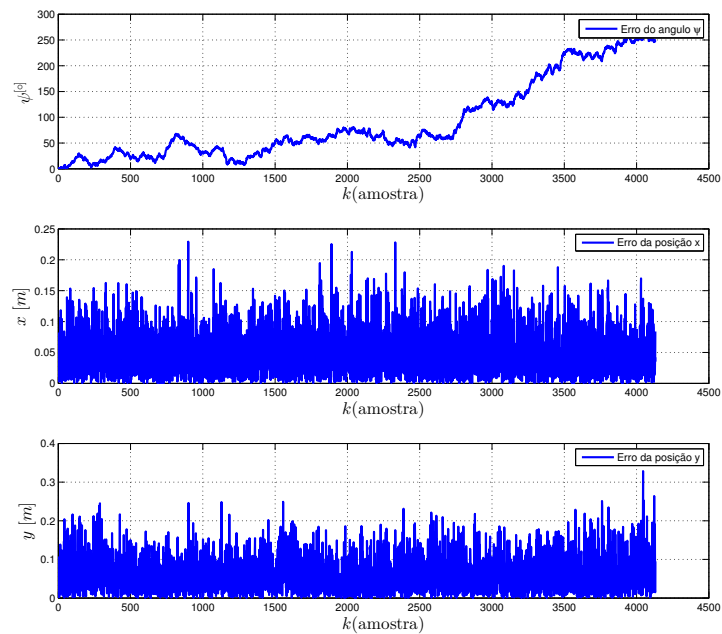
Os resultados obtidos para primeira trajetória utilizando o FKE com três boias estão apresentados na Figura 31. O filtro foi capaz de determinar a posição em  $x$  e  $y$  do veículo com um erro de estimação entre 0.2 m como é mostrado na Figura 32.

Figura 31 – Resultado do FKE para a primeira trajetória utilizando três boias



Fonte: o autor, (2016)

Figura 32 – Erro da estimação obtido utilizando o FKE com três boias para primeira trajetória

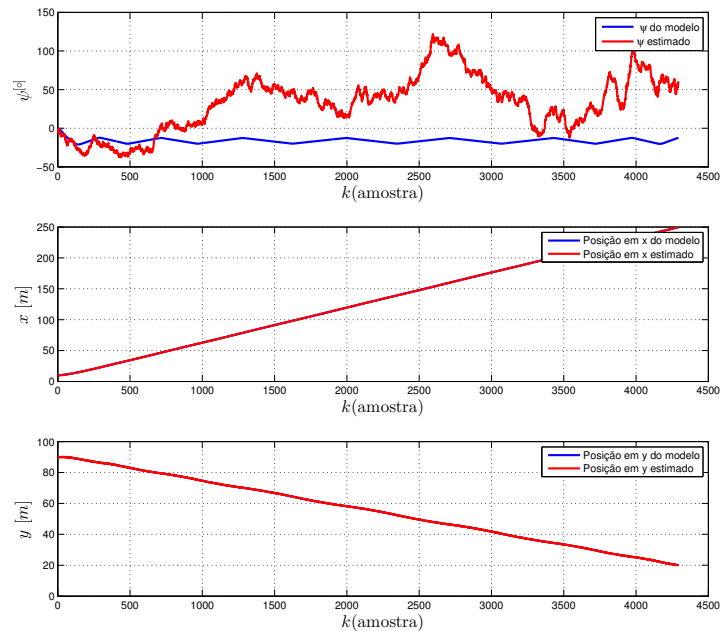


Fonte: o autor, (2016)

Para segunda trajetória os resultados obtidos na estimação da pose  $p_k = [x \ y \ \psi]$  do veículo são apresentados na Figura 33. Neste caso o filtro foi capaz de determinar a

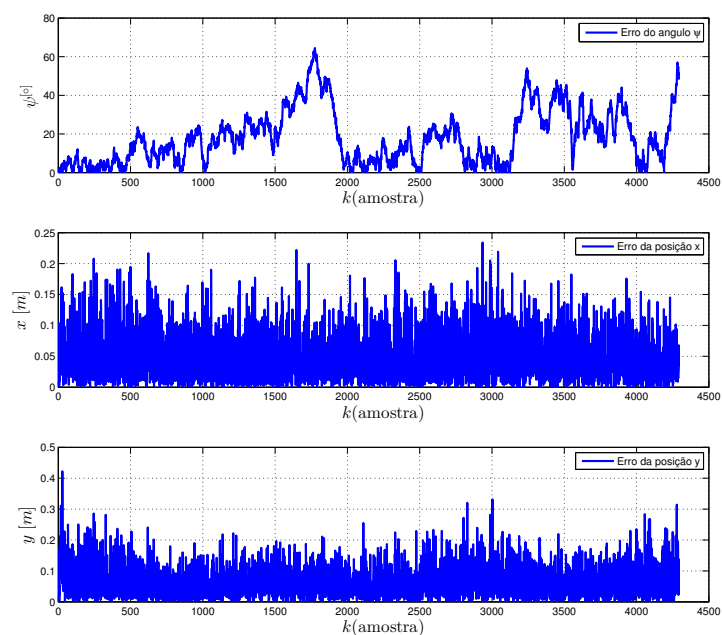
posição  $x$  e  $y$  do veículo com um erro de estimação de aproximadamente 0.2 m, como é mostrado no gráfico do erro apresentado na Figura 34. Com o uso de três boias e para essa trajetória o filtro não foi capaz de estimar o ângulo de orientação  $\psi$  do sistema.

Figura 33 – Resultado do FKE para a segunda trajetória utilizando três boias



Fonte: o autor, (2016)

Figura 34 – Erro da estimação obtido com o FKE para a segunda trajetória utilizando três boias

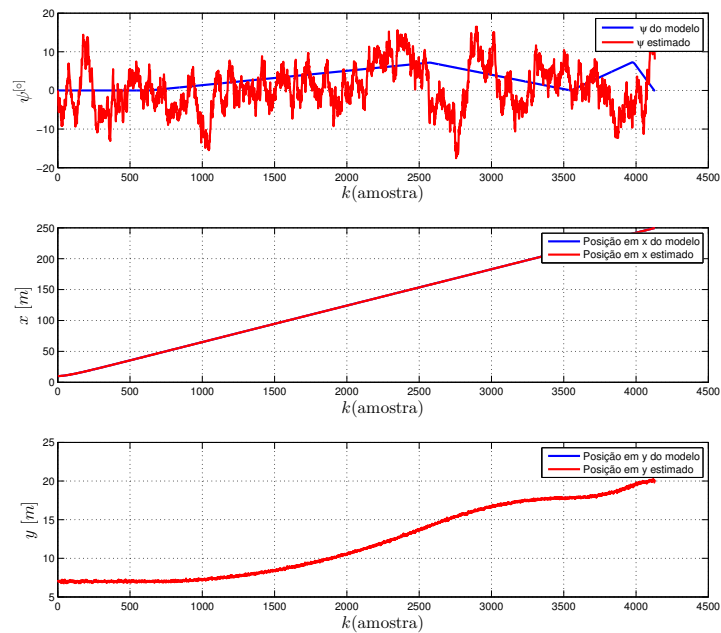


Fonte: o autor, (2016)

### 3.3.4 Resultados obtido com três boia e o sensor virtual

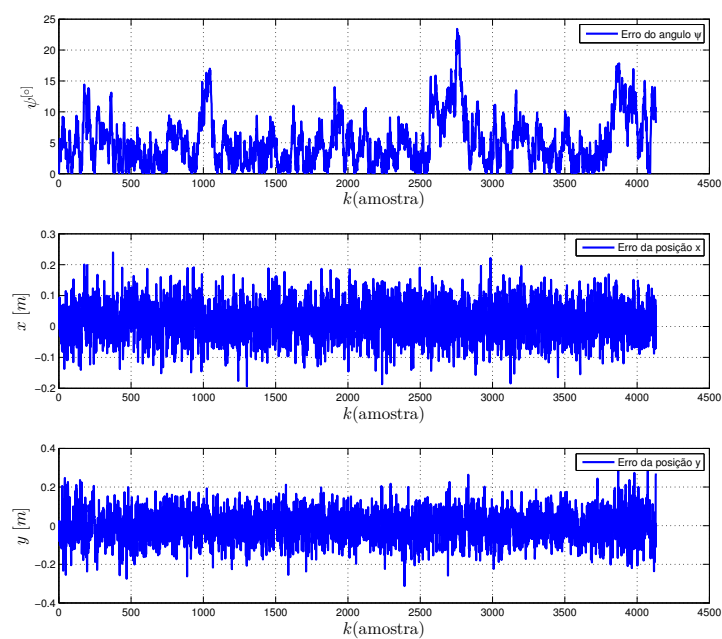
Para solucionar o problema de estimação do ângulo  $\psi$  foi desenvolvido um sensor virtual que foi descrito na seção 3.2. Foram realizadas simulações usando três boias localizadas na mesma posição do método anterior (Figura 30), e para os valores utilizados para a matriz de incertezas dos sensores foi de  $Q_k = \text{diag} \{[10^{-3} \ 10^{-3} \ 10^{-3} \ 5 \ 5]\}$ . Os resultados obtidos da estimação da pose  $p_k = [x \ y \ \psi]$  do veículo utilizando o FKE com três boias e com o sensor virtual para primeira trajetória são apresentados na Figura 35. Pode-se ver que a determinação do ângulo  $\psi$  melhorou em relação as outras técnicas apresentadas, onde o erro de estimação ficou próximo de zero em alguns instantes da simulação é mostrado na Figura 38.

Figura 35 – Resultado do FKE para primeira trajetória utilizando três boias e com sensor virtual



Fonte: o autor, (2016)

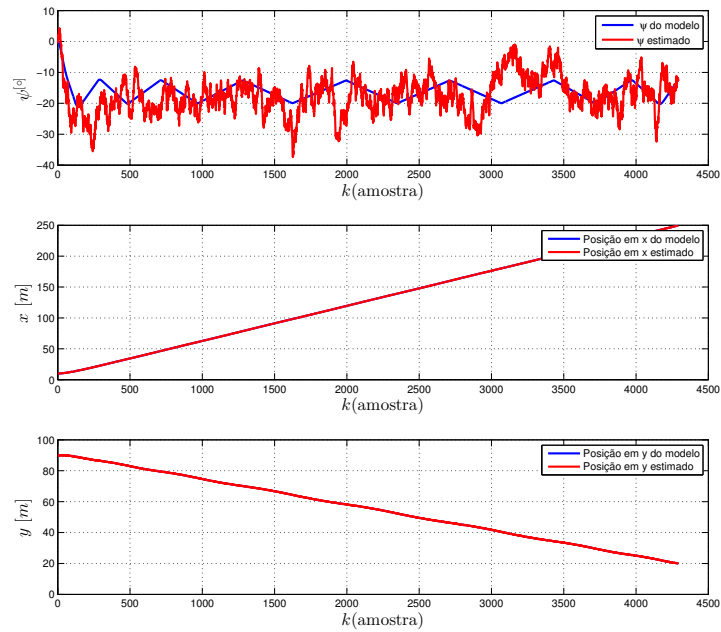
Figura 36 – Erro da estimação obtido com o FKE para primeira trajetória utilizando três boias e com sensor virtual



Fonte: o autor, (2016)

Para segunda trajetória os resultados da estimação dos estados são apresentados na Figura 37, onde pode-se ver a melhora na estimação do ângulo  $\psi$ . Na Figura 38 é possível ver que o erro de estimação de  $\psi$  está oscilando em zero, o que para as outras técnicas não era possível.

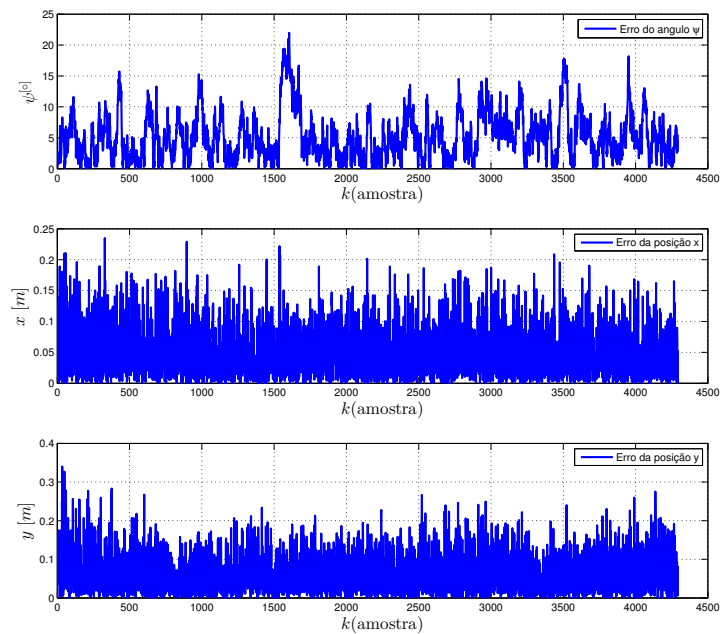
Figura 37 – Resultado do FKE para segunda trajetória utilizando três boias e com sensor virtual



Fonte: o autor, (2016)



Figura 38 – Erro da estimação obtido com o FKE para a segunda trajetória utilizando três boias e com sensor virtual



Fonte: o autor, (2016)

### 3.4 Análise de Desempenho

Para realizar a análise de desempenho da técnica de localização baseado no FKE foi realizado o cálculo do erro euclidiano da trajetória estimada pelo filtro. A média desse erro está demonstrado na tabela 2 para primeira, e segunda trajetória. Logo analisando esse resultados é possível afirmar que é inviável utilizar apenas uma boia para determinar a localização de um veículo aquático utilizando o FKE. A técnica de localização baseada no FKE com duas boias apresentou um resultado com erro médio de estimação pequeno para primeira trajetória, já para a segunda esse erro ficou um pouco maior. O melhor desempenho apresentado foi com três boias, onde tanto para primeira e segunda trajetória obteve-se um erro médio de estimação pequeno.

Tabela 2 – Tabela com o erro de estimação das trajetória para o método de localização baseado no FKE.

Método	Erro (primeira trajetória)	Erro (segunda trajetória)
FKE com uma boia	58 m	54,5 m
FKE com duas boias	0.09 m	0.21 m
FKE com três boias	0.08 m	0.09 m
FKE com três boias e sensor virtual	0.08 m	0.09 m

Fonte: o autor, (2016)

Foi realizado o calculo do erro da trajetória euclidiana com a seguinte expressão:

$$e_T = \frac{1}{N} \sum_i^N \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \quad (3.15)$$

onde,  $N$  representa o número de amostras,  $\hat{x}$  e  $\hat{y}$  o estado estimado pelo filtro,  $x$  e  $y$  o valor real oriundo do modelo do veículo.

A análise do ângulo de orientação do veículo  $\psi$  foi feita através do calculo do erro médio, e os resultados obtidos para cada abordagem estão descritos na tabela 3. Pode-se ver na tabela que a única técnica que apresentou um erro de estimação pequeno para o estado  $\psi$  foi o FKE com três boias em conjunto com o sensor virtual. As outras técnicas que não foram utilizadas em conjunto com o sensor virtual apresentaram um erro maior que  $10^\circ$ , onde na prática pode ser um valor inviável de tolerar.

Tabela 3 – Tabela com o erro médio de estimação do ângulo  $\psi$  para o método de localização baseado no FKE.

Método	Erro (primeira trajetória)	Erro (segunda trajetória)
FKE com uma boia	17.09°	138.7°
FKE com duas boias	75.07°	34.89°
FKE com três boias	34.8°	44.3°
FKE com três boias e sensor virtual	5.08°	4.8°

Fonte: o autor, (2016)

O calculo do erro de estimação do ângulo  $\psi$  foi feito da seguinte maneira:

$$e_\psi = \frac{1}{N} \sum_i^N \sqrt{(\psi_i - \hat{\psi}_i)^2}, \quad (3.16)$$

tal que,  $N$  representa o número de amostras,  $\hat{\psi}$  o estado estimado pelo filtro e  $\psi$  o valor real oriundo do modelo do veículo.

### 3.5 Comentários Finais

Neste capítulo foi apresentado um método de localização para um veículo aquático baseado no Filtro de Kalman Estendido utilizando boias. Para validar a técnica apresentada foram realizadas simulações utilizando o *software MATLAB* com diferentes quantidades de boias. Assim foi constatado que para obter uma estimação com um bom desempenho no mínimo é necessário utilizar duas boias. Para melhorar o desempenho do filtro foi desenvolvido um sensor virtual para corrigir a estimação do ângulo  $\psi$ . Uma análise do erro média da trajetória estimada para cada abordagem foi realizada, e também do erro de estimação do ângulo de orientação.

Entretanto, apesar do bom desempenho esta técnica está relacionada com o seu custo de implementação na pratica que é alto, devido a utilização das boias. Por esse motivo no próximo capítulo são apresentadas soluções que possuem um menor custo se comparado com a técnica apresentada, e apresentaram um desempenho satisfatório.

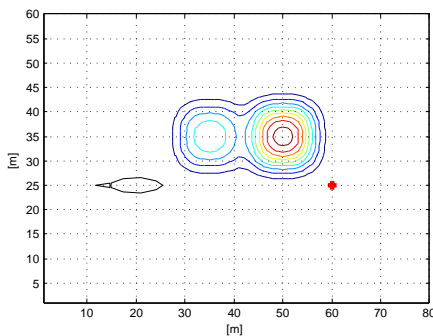
## 4 Localização Baseada no Filtro de Partículas

Nesta capítulo será apresentado a implementação do Filtro de Partículas para determinar a localização do veículo. Foram implementadas duas técnicas diferentes, a Localização Baseada em Terreno (LBT) e a baseada em sensores acústicos (sistema GIB). Para avaliar o desempenho dos métodos desenvolvidos simulações foram realizadas no software *MATLAB* e esses resultados são descritos na última seção deste capítulo.

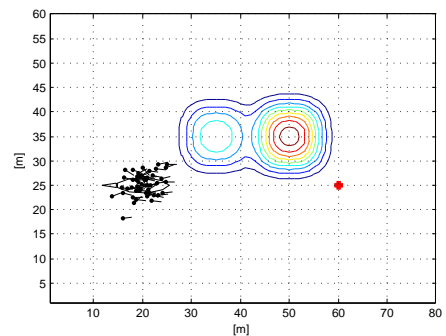
### 4.1 Predição

No primeiro passo da predição é inicializado um grupo  $\chi_k^M = \{\bar{p}_k^1, \bar{p}_k^2, \bar{p}_k^3, \dots, \bar{p}_k^M\}$  de  $M$  partículas de forma aleatória ao redor da posição inicial do veículo (Figura 39), supõem-se que a posição inicial do veículo foi obtida através de um GPS antes da submersão. Assim cada  $\bar{p}_k^M$  representa uma predição da *pose*  $\bar{p}_k^M = [\bar{x}^i, \bar{y}^i, \bar{\psi}^i]$  do veículo.

Figura 39 – Primeiro passo da predição. A) Veículo na posição inicial sem a inicialização das partículas. B) Inicialização da partículas em torno da posição inicial do veículo



[A]



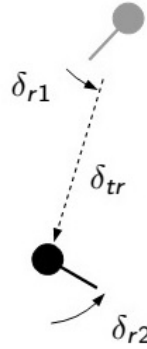
[B]

Fonte: o autor, (2016)

No próximo passo o veículo submerge e começa a se movimentar, nesse momento as partículas também começam a se mover, mas com uma dinâmica própria. O deslocamento realizado por cada partícula é governado pelo modelo RTR (Rotação, Translação, Rotação). Esse modelo calcula a nova estimativa da *pose* de uma partícula através

de uma rotação  $\delta_{r1}$ , seguida de uma translação  $\delta_{t1}$  e uma segunda rotação  $\delta_{r2}$  como é mostrado na Figura 40.

Figura 40 – Ilustração do modelo RTR: Rotação inicial  $\delta_{r1}$ , translação  $\delta_{t1}$  e rotação final  $\delta_{r2}$



Fonte: THRUN; BURGARD; FOX, (2005)

Na Figura 41 é descrito o algoritmo do modelo RTR (THRUN; BURGARD; FOX, 2005), as medições do IMU são adicionadas a predição da pose  $\bar{p}_k$  do sistema. Esse algoritmo recebe como entrada  $\Delta_x$ ,  $\Delta_y$  e  $\Delta_\psi$  que são as medições que foram obtidos pela integração das medidas do IMU são determinados pela seguinte expressão:

$$\begin{aligned}\Delta_x &= \iint_{k-1}^k a_u \\ \Delta_y &= \iint_{k-1}^k a_v \\ \Delta_\psi &= \iint_{k-1}^k g_\psi\end{aligned}\quad (4.1)$$

Esse algoritmo também recebe os valores estimados da *pose* anteriores  $p_k^M = [\bar{x}_{k-1}^M \quad \bar{y}_{k-1}^M \quad \bar{\psi}_{k-1}^M]'$ , e retorna uma nova pose “perturbada”. As três primeiras linhas (1 até 3) do algoritmo são responsáveis pela estimativa da rotação inicial, translação e rotação final que o veículo teria realizado (Figura 40). Nas linhas 5 até 7 são calculadas variações desses movimentos baseados em números aleatórios com média  $\mu$  e covariância  $\sigma^2$  e que são governados pelos parâmetros  $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ . Finalmente nas linhas 8 até 10 são calculadas novas estimativas da pose  $\bar{p}_k^M$ , essas duas últimas etapas são feitas para cada uma das partículas ( $i = 1 \dots M$ ). Na linha 11 o algoritmo retorna uma predição a priori da *pose* do veículo.

Figura 41 – Algoritmo do modelo RTR.

---

```

1: procedure MODELORTR( $\Delta_x, \Delta_y, \Delta_\psi, \bar{p}_{k-1}^M = [\bar{x}_{k-1}^M \ \bar{y}_{k-1}^M \ \bar{\psi}_{k-1}^M]'$ )
2:    $\delta_{r1} = \text{atan2}(\Delta_y, \Delta_x) - \Delta_\psi$ 
3:    $\delta_{tr} = \sqrt{(\Delta_x)^2 + (\Delta_y)^2}$ 
4:    $\delta_{r2} = \Delta_\psi - \delta_{r1}$ 

5:   for  $i=1$  to  $M$  do
6:      $\bar{\delta}_{r1} = \text{rand}(\mathcal{N}(\delta_{r1}, \alpha_1\delta_{r1}^2 + \alpha_2\delta_{tr}^2))$ 
7:      $\bar{\delta}_{tr} = \text{rand}(\mathcal{N}(\delta_{tr}, \alpha_3\delta_{tr}^2 + \alpha_4\delta_{r1}^2 + \alpha_4\delta_{r2}^2))$ 
8:      $\bar{\delta}_{r2} = \text{rand}(\mathcal{N}(\delta_{r2}, \alpha_1\delta_{r2}^2 + \alpha_2\delta_{tr}^2))$ 

9:      $\bar{x}_k^i = \bar{x}_{k-1}^i + \bar{\delta}_{tr}\cos(\bar{\psi}_{k-1}^i + \bar{\delta}_{r1})$ 
10:     $\bar{y}_k^i = \bar{y}_{k-1}^i + \bar{\delta}_{tr}\text{sen}(\bar{\psi}_{k-1}^i + \bar{\delta}_{r1})$ 
11:     $\bar{\psi}_k^i = \bar{\delta}_{r1} + \bar{\delta}_{r2}$ 
12:    return  $\hat{p}_k^i = [\bar{x}_k^i \ \bar{y}_k^i \ \bar{\psi}_k^i]'$ 

13:   end for
14: end procedure

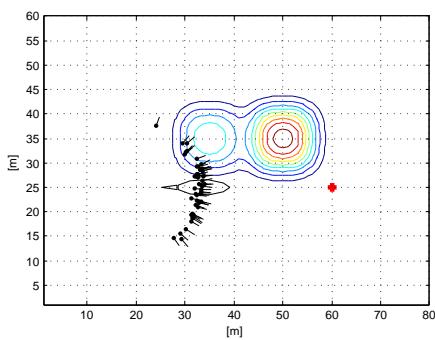
```

---

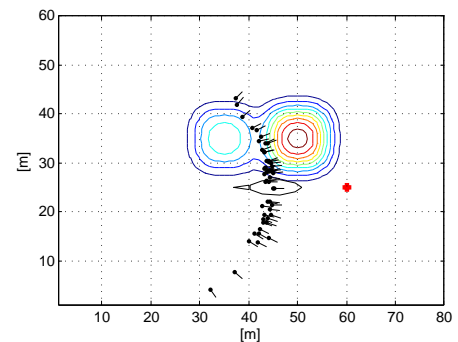
Fonte: o autor, (2016).

Na Figura 42 mostra-se como funciona o algoritmo RTR, onde as partículas que foram inicialmente criadas ao redor da posição do veículo começam a se locomover governadas por  $\delta_{r1}$ ,  $\delta_{tr}$  e  $\delta_{r2}$ . Esse movimento das partículas é feito até ocorrer o passo de correção.

Figura 42 – Segundo passo da predição. A) movimentação das partículas pelo RTR B) as partículas continuam a se mover governadas pelo RTR até ocorrer a correção



[A]

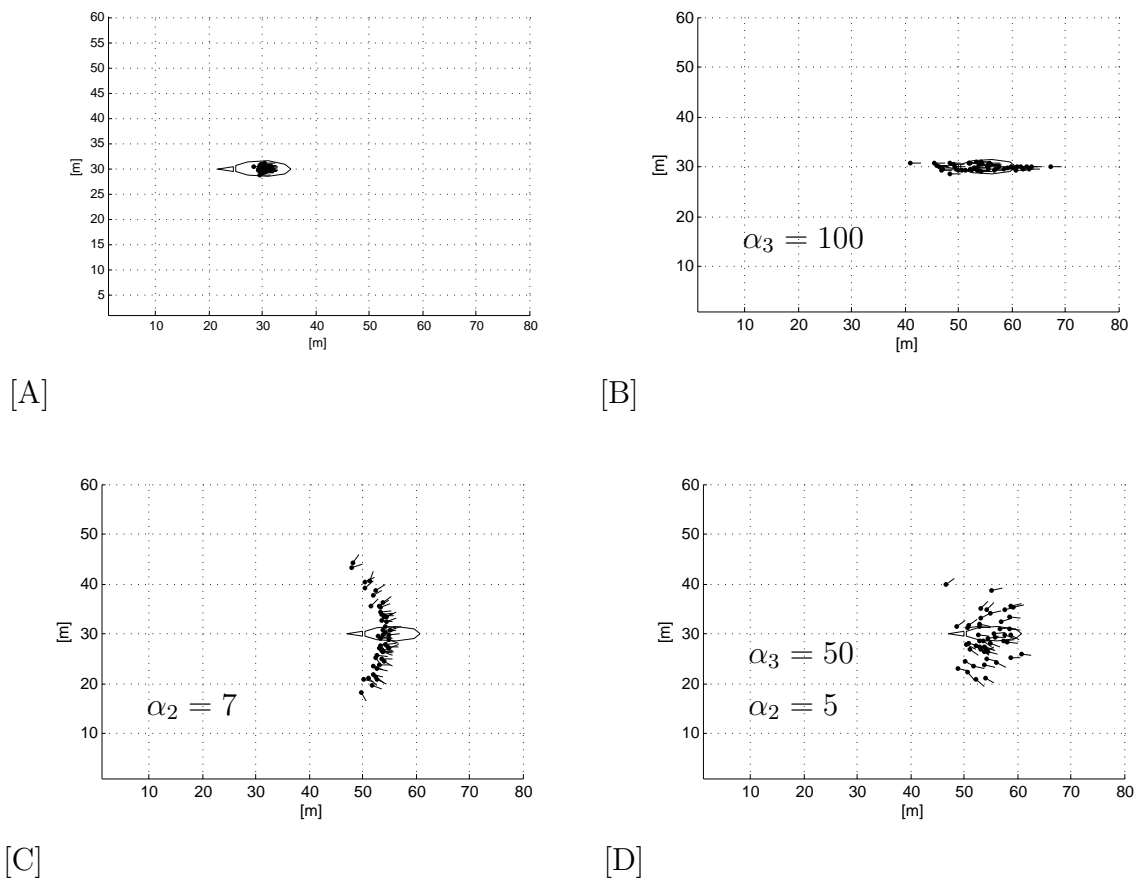


[B]

Fonte: o autor, (2016)

Os parâmetros  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$  e  $\alpha_4$  são responsáveis pela variabilidade das partículas (Figura 43). Por exemplo na Figura 43 A todos os valores estão em  $\alpha = 0$ , nota-se que não ocorre nenhuma variabilidade, já na Figura 43 B as partículas apenas estão variando no eixo  $x$  isso porque o  $\alpha_3 = 100$  e os outros estão em zero. Com  $\alpha_2 = 7$  e os outros com valores iguais a zero as partículas apenas se dispersaram no eixo  $y$  (Figura 43 C), já na Figura 43 D as partículas se dispersaram em todos os eixos.

Figura 43 – Efeitos dos  $\alpha$  A) Todos os  $\alpha = 0$ . B) Todos os  $\alpha = 0$  exceto o  $\alpha_3 = 100$ . C) Todos os  $\alpha = 0$  exceto o  $\alpha_2 = 7$ . D) Mais de um  $\alpha$  diferente de zero.



Fonte: o autor, (2016)

## 4.2 Correção Baseada em Boias

O passo de correção baseado em boias implementado funciona em duas etapas, na primeira etapa é atribuído um fator de importância  $w_k^M$  para cada partícula que diz o quanto a estimativa da *pose* está correta em relação a posição real. Para isso é realizada a medição da distancia até a boias  $\hat{r}^i$  para cada partícula através da equação (2.22) utilizando os valores preditos de  $\hat{p}_k^M$  pelo algoritmo 41. Dessa forma é atribuído para cada

partícula um fator de importância que é calculado através de uma distribuição normal, assim  $w_k^M$  é determinado pela seguinte equação:

$$w_k^M = \frac{1}{\sigma_w \sqrt{2\pi}} e^{-\frac{(\hat{r}_k^M - r_k)^2}{2\sigma_w^2}} w_{k-1}^M \quad (4.2)$$

Na Figura 44 descreve como foi implementado o algoritmo para realizar a atribuição de pesos para cada partícula (THRUN; BURGARD; FOX, 2005), e recebe como entrada os valores de pesos  $w_{k-1}^M$  do instante passado, a medição real da distancia da boias  $r_i$ , a predição da *pose*  $p_k^M$ , a profundidade que o veículo está navegando  $d$ , o ruído das boias  $\varepsilon_r$  e as posições das boias  $\mathbf{m}_{j,x}$   $\mathbf{m}_{j,y}$ , onde  $M$  representa o número de partículas e  $i_{max}$  o número de bois. Na linha 4 é calculado a distância virtual de cada partícula até cada boia considerada. Com a distância real de cada boia  $r^i$  e a distância virtual de cada partícula, na linha 5 é calculado um peso de importância  $wr_i^j$  para cada boia considerada. Por ultimo na linha 7 determina-se o peso final  $w_k^j$  para cada partícula multiplicando o peso anterior  $w_{k-1}^j$  e os pesos calculados para cada boia considerada ( $wr_{i=1}^j wr_{i=2}^j \dots wr_{i_{max}}^j$ ). Na linha 9 retorna-se o peso de importância final  $w_k^M$  calculado para todas as partículas.

Figura 44 – Algoritmo de atribuição dos pesos  $w$  para cada partícula.

---

```

1: procedure ATRIBUIÇÃO_PESOS( $w_{k-1}^M, r^i, \bar{p}_k^M = [\bar{x}_k^M \ \bar{y}_k^M \ \bar{\psi}_k^M]'$ ,  $d, \varepsilon_r, \mathbf{m}_{j,x}, \mathbf{m}_{j,y}$ )
2:   for  $j = 1$  to  $M$  do
3:     for  $i = 1$  to  $i_{max}$  do
4:        $\hat{r}_i^j = \sqrt{(\mathbf{m}_{i,x} - \bar{x}_k^j)^2 + (\mathbf{m}_{i,y} - \bar{y}_k^j)^2 + d^2 + \varepsilon_r}$ 
5:        $wr_i^j = \frac{1}{\sigma_w \sqrt{2\pi}} e^{-\frac{(\hat{r}_i^j - r_i)^2}{2\sigma_w^2}} w_{k-1}^M$ 
6:     end for
7:      $w_k^j = w_{k-1}^j (wr_{i=1}^j wr_{i=2}^j \dots wr_{i_{max}}^j)$ 
8:   end for
9:   return  $w_k^M$ 
10: end procedure

```

---

Fonte: o autor, (2016).

Após cada partícula obter um peso de importância  $w_k^M$  é feito a escolha das melhores partículas para compor a estimação final da *pose* do sistema. O método utilizado para isso é o *Importance Sampling*, o qual faz com que as partículas com alta importância so-



brevivam e se reproduzam. Ou seja as partículas com menor importância são substituídas por cópias das partículas com alto fator de importância (HASTINGS, 1970).

Um método simples de implementar o método de *Importance Sampling* e através do algoritmo 45 que é o *Sampling Wheel* (THRUN; BURGARD; FOX, 2005), o qual realiza a seleção das partículas e recebe como entrada o vetor de pesos  $w_k^M$  e a predição da pose do sistema  $\bar{p}_k^M$ . A seleção das melhores partícula é realizada com uma determinada frequência *ResampleTime* que é um parâmetro ajustável, se esse passo é feito sempre, as partículas não conseguem se propagar e se diferenciar entre si. Na segunda linha é criado uma variável  $\beta$  que serve como um limiar variante de escolha. As partículas com probabilidade  $w_j > \beta, j = 1 \dots M$  serão selecionadas e irão compor o próximo conjunto de partículas, ou seja são inseridas em  $\chi$ . Na terceira linha uma variável  $j$  é criada para percorrer o vetor de pesos  $w = [w_k^1, w_k^2, \dots, w_k^M]$ , a qual é criada de uma forma aleatória. Um laço de repetição é feito na quarta linha para selecionar  $M$  partículas e inseri-las no vetor  $\chi$ . Para isso o limiar  $\beta$  também é inicializado de forma aleatória e comparado com os valores de  $w^j$  das partículas. Assim quando um valor de  $w^j$  for maior que  $\beta$ , a partícula atrelada a esse peso é selecionado para compor  $\chi$ . Do contrário o limiar  $\beta$  é reduzido e a comparação é feita com o próximo valor do vetor  $w$ . Dessa forma existe uma pequena probabilidade de que partículas com baixo  $w$  sejam selecionadas. Na linha 14 todas as partículas selecionadas no vetor  $\chi$  são atribuídas ao vetor  $\hat{p}_k$  formando assim a nova estimativa da pose do sistema.

Dessa forma algumas partículas serão selecionadas e reproduzidas inúmeras vezes, enquanto que outras não serão e morrerão. As partículas que foram copiadas irão se diferenciar através do próximo passo de predição pelo modelo RTR e também porque o passo de correção não é feito em todo o instante de tempo  $k$ . Para evitar que sempre a mesma partícula seja selecionada o passo inicial do algoritmo de correção começa em um ponto aleatório do vetor  $w$ , como mostra a linha 3 do algoritmo 45.

Figura 45 – Algoritmo de seleção das melhores partículas.

---

```

1: procedure SAMPLINGWHEEL( $w_k^M = [w_k^1, w_k^2, \dots, w_k^M]$ ,  $\bar{p}_k^M = [\bar{x}_k^M \ \bar{y}_k^M \ \bar{\psi}_k^M]'$ )
2:   if  $k == ResampleTime$  then
3:      $\beta = 0$ ;
4:      $j = \text{int}(\text{rand}([1 \ M]))$ 
5:     for  $i=1$  to  $M$  do
6:        $\beta = \beta + \text{rand}([0 \ \max(w)])$ 
7:       while  $\beta > w^j$  do
8:          $\beta = \beta - w^j$ 
9:          $j = j + 1$ 
10:        if  $j > M$  then
11:           $j = 1$ 
12:        end if
13:      end while
14:       $\chi_i = \hat{p}_j$ 
15:    end for
16:     $\hat{p}_k = \chi$ 
17:  end if
18:  return  $\hat{p}_k$ 
19: end procedure

```

---

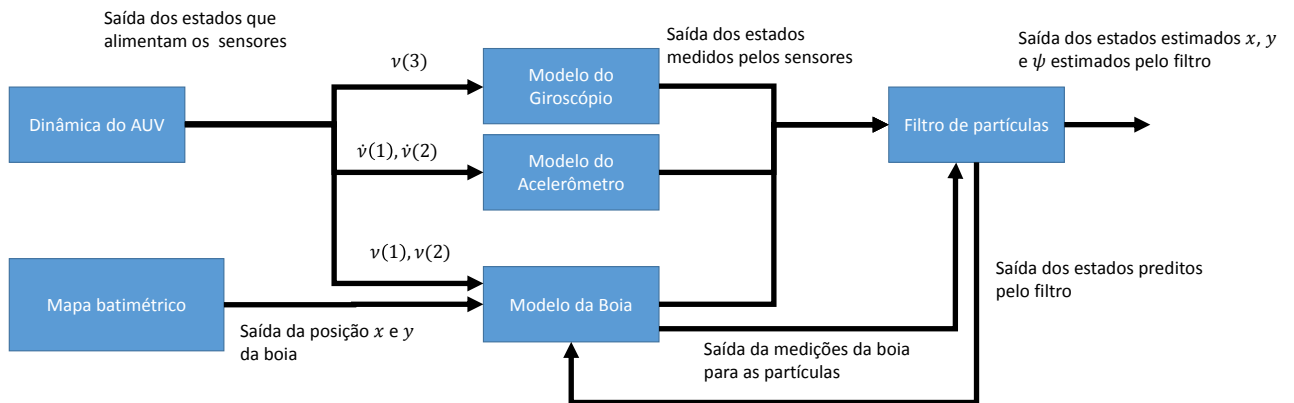
Fonte: o autor, (2016).

No final deste passo é calculada a estimativa final da *pose*  $p_k$  do veículo através da média de todas partículas no instante de tempo  $k$  da seguinte forma:

$$\tilde{\mathbf{x}}_k = \frac{1}{M} \sum_{i=1}^M \hat{p}_k \quad (4.3)$$

O método de localização baseado no Filtro de Partículas utilizando boias desenvolvido neste trabalho está descrito no diagrama da Figura 46. Para o passo de predição desta técnica foi utilizado a *IMU* (acelerômetro e giroscópio) e o algoritmo do modelo RTR apresentado. Na etapa de correção foi utilizado o sistema *GIB* (boia), desse modo é feito o cálculo da distância  $\hat{r}$  para cada partícula até a boia utilizando a expressão (2.22), e o mapa foi utilizado para determinar a posição das boias.

Figura 46 – Diagrama de blocos da técnica de localização baseado no FP utilizando boias

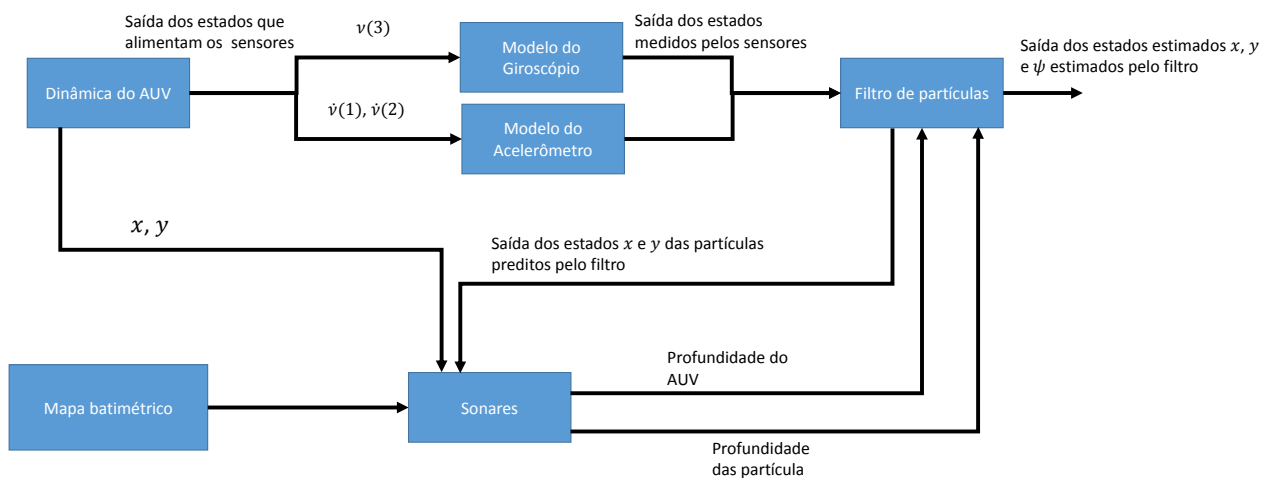


Fonte: o autor, (2016)

### 4.3 Localização Baseada em Terreno

O método de localização baseado em terreno desenvolvido neste trabalho utiliza o Filtro de Partículas para realizar a fusão dos dados de um sensor IMU, sonar e um mapa batimétrico (mapa com informação do terreno). Na figura 47 é apresentado o diagrama de blocos da técnica LBT implementada, onde o modelo do sistema é utilizado apenas para simular os sensores, dessa forma o FP fica totalmente desacoplado do modelo do sistema. Como foi descrito anteriormente o filtro funciona em duas etapas, uma onde ocorre a predição das medidas, e na outra a correção.

Figura 47 – Diagrama do Filtro de Partículas baseado em terreno.



Fonte: o autor, (2016)

O método de Localização Baseado em Terreno realiza o passo de predição com os mesmos sensores e algoritmos da técnica baseada em boias. Mas realiza a atribuição dos pesos  $w_k^M$  de maneira diferente, isso porque é nessa etapa que o filtro adiciona a informação do mapa batimétrico. Para isso é realizado a medição  $\hat{z}_k^M$  para cada partícula através da equação (2.21) utilizando os valores preditos de  $\hat{p}_k^M$ .

Figura 48 – Algoritmo de atribuição dos pesos  $w$  para cada partícula.

---

```

1: procedure ATRIBUIÇÃO_PESOS( $w_{k-1}^j, z_k^i, M_{mapa}(x, y), \bar{p}_k^M = [\bar{x}_k^M \ \bar{y}_k^M \ \bar{\psi}_k^M]'$ ,  $d, \varepsilon_z$ )
2:   for  $j=1$  to  $M$  do
3:     if  $[\bar{x}_k^j] > size(M_{mapa}, x)$  then
4:        $w_k^j = 0$ 
5:     else if  $[\bar{y}_k^j] > size(M_{mapa}, y)$  then
6:        $w_k^j = 0$ 
7:     else
8:        $\hat{z}_k^j = M_{mapa}(\bar{x}_k^j, \bar{y}_k^j) - d + \varepsilon_z$ 
9:          $wp_k^j = \frac{1}{\sigma_w \sqrt{2\pi}} e^{-\frac{(\hat{z}_k^j - z_k)^2}{2\sigma_w^2}}$ 
10:       $w_k^j = w_{k-1}^j wp_k^j$ 
11:    end if
12:  end for
13:  return  $w_k^M$ 
14: end procedure

```

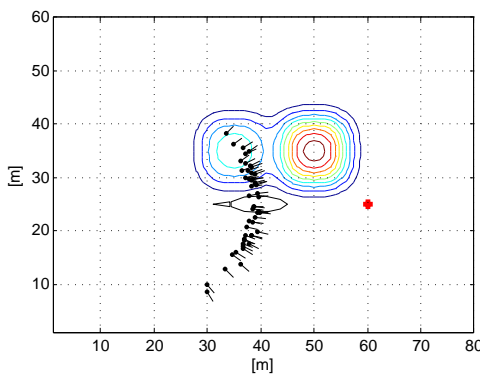
---

Fonte: o autor, (2016).

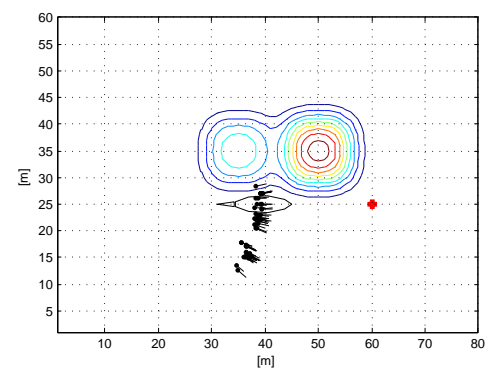
Dessa forma é atribuído para cada partícula um peso de importância  $w_k^M$  através do algoritmo descrito na Figura 48, o qual recebe como entrada a medição real do sonar  $Z_k$ , a pose predita  $\bar{p}_k^M$ , os valores de peso no instante passado  $w_{k-1}^M$ , a matriz que representa o mapa batimétrico  $M_{mapa}(x, y)$ , a profundidade  $d$  que o veículo está navegando e o ruído do sonar  $\varepsilon_z$ . Para as partículas que se encontram fora do mapa é atribuído peso  $w_k^M = 0$  isso é feito entre as linhas 3 e 7. Na linha 8 para cada uma das partículas é calculado a medição do sonar virtual  $\hat{z}_k^M$ , através da expressão (2.21), depois na linha 9 é determinado o peso de importância  $wp_k^M$  para cada partícula. Assim por último na linha 10 calcula-se o peso final  $w_k^M$  para cada partícula multiplicando o peso de importância anterior  $w_{k-1}^M$  pelo peso de cada partícula  $wp_k^M$ . No final (linha 13) desse algoritmo é retornado o peso de importância  $w_k^M$  calculado para todas as partículas.

Depois de realizar o passo de atribuição de pesos é utilizado o Algoritmo 45 de seleção das melhores partículas. Na Figura 69 A mostra as partículas se dispersando-se governadas pelo modelo RTR, onde muitas estão na posição errada. Já na Figura 69 com a ação do passo de correção as partículas que estavam na posição errada foram substituídas por cópias das partículas que estavam corretas.

Figura 49 – Exemplo do passo de correção do método LBT A) Partículas se dispersando governadas pelo modelo RTR. B) Ação do passo de correção, onde as partículas com baixa probabilidade foram eliminadas.



[A]



[B]

Fonte: o autor, (2016)

No final do passo de correção a estimativa da *pose*  $p_k$  do veículo é realizada através, do calculo da média de todas as partículas no instante de tempo  $k$ ,

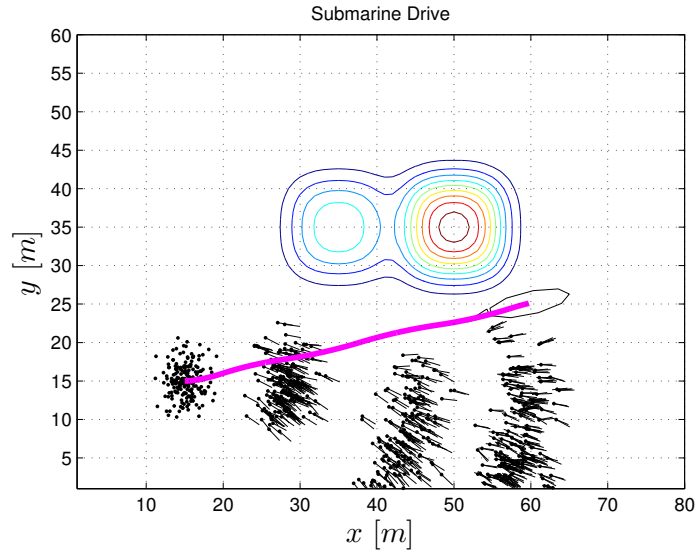
$$\tilde{\mathbf{x}}_k = \frac{1}{M} \sum_{i=1}^M \hat{p}_k^i$$

#### 4.4 Deficiência da Localização Baseada em terreno

A boa performance do Filtro de Partículas depende da etapa em que o fator de importância  $w_k^i$  é calculado. Nesse passo o filtro compara a profundidade virtual  $\hat{z}_k^i$  obtida através do mapa, com a profundidade real  $z_k$  do veículo adquirido pelo sonar. Dessa forma, é importante salientar que todas as partículas que estiverem em uma região do mapa com a mesma profundidade irão possuir o mesmo valor de  $\hat{z}_k^i$  e conseqüentemente o mesmo  $w_k^i$ .

Para diferenciar as partículas a única maneira e se em instantes passados, as mesmas passarem por regiões do mapa com diferentes profundidades. Dessa forma os pesos  $w_{k-l}^i$  para  $l > 1$ , serão diferentes de  $w_{k-l}^j$  para  $j \neq i$ . No entanto, se o terreno não possuir suficiente variação de profundidade, em algum momento as partículas não conseguiram se diferenciar, levando o filtro a estimar a *pose* do sistema de forma errada (Figura 50).

Figura 50 – Problema da Localização Baseada em terreno, as partículas não conseguem se diferenciar em regiões com profundidade igual. A linha roxa mostra a trajetória real do veículo.



Fonte: o autor, (2016)

## 4.5 Método de Geração de Trajetória

Para contornar o problema explicado acima um algoritmo de geração de trajetória foi implementado. A técnica empregada considerada a variação de profundidade do mapa para gerar um caminho que evita regiões planas, isso para garantir um trajeto mais seguro para o método de localização do veículo. O método utilizado define uma ação ótima  $\delta$  para cada estado  $(x, y)$  baseado em MDPs (*Markov Decision Process*). A *policy*  $\pi_{pol}$  representa a melhor ação para cada estado e cada *policy* possui uma função *value*  $V$  associada, que representa o *cumulative payoff* descontado pelo fator  $\gamma$ . Mas, uma função de *local payoff*  $R_{pay}(x, y)$  que determina o custo de movimento para cada posição do mapa é necessário para determinar a função  $V$ , esse método de planejamento de trajetória foi apresentado em (THRUN; BURGARD; FOX, 2005). A fim de determinar tal função, primeiro calcula-se a maior variação de terreno em cada coordenada:

$$M_I(x, y) = \max(|\nabla M_{mapa}(x)|, |\nabla M_{mapa}(y)|) \quad (4.4)$$

onde  $M_{mapa}$  representa uma matriz  $2D$  contendo as coordenadas  $x$  e  $y$  do ambiente e suas respectivas profundidades  $z$ . Além disso,  $\nabla M_{mapa}(i), i = x, y$  representa o gradiente do mapa na direção  $i$  e  $M_I(x, y)$  é a matriz de variação do terreno que é normalizada, então possui valores entre  $[0, 1]$  (valores mais próximos de um representa regiões com maior variação de terreno). O algoritmo descrito na Figura 51 determina a matriz  $M_I$ ,

na linha 1 é calculado o gradiente do mapa nas duas coordenadas  $F_x$  e  $F_y$ . Depois entre as linhas 2 e 6 determina-se o maior valor de gradiente entre as coordenadas. Por último na linha 7 é feita a normalização da matriz  $M_I$ .

Figura 51 – Algoritmo do calculo da variação do terreno.

---

```

1: procedure VARIAÇÃO TERRENO(  $M_{mapa}(x, y)$ )
2:    $[F_x, F_y] = gradient(M_{mapa}(x, y))$ 
3:   for  $i=1$  to size( $F_x$ ) do
4:     for  $j=1$  to size( $F_y$ ) do
5:        $M_I(i, j) = max(abs(F_x(i, j)), abs(F_y(i, j)))$ ;
6:     end for
7:   end for
8:    $M_I = M_I / max(max(M_I))$ ;
9:   return  $M_I$ 
10: end procedure

```

---

Fonte: o autor, (2016).

Para cada posição um custo de movimento  $R_{pay}(x, y)$  é determinado pelo algoritmo descrito na Figura 52, assim  $x_g$  e  $y_g$  representam a posição de chegada com um respectivo custo dado por  $\lambda_g \gg 1$ . O custo de se movimentar para qualquer outra coordenada do mapa é denotado por  $0 < \lambda_c \leq 1$ . Na linha 6 o algoritmo evita oscilações na trajetória, considerando a importância do terreno apenas quando um limiar  $0 < \Upsilon < 1$  é ultrapassado.

Figura 52 – Algoritmo da *Payoff Function*


---

```

1: procedure  $M_I(x, y)$ 
2:   for all  $x$  do
3:     for all  $y$  do
4:       if  $x=x_g \wedge y=y_g$  then
5:          $R_{pay}(x, y)=\lambda_G$ 
6:       else if  $M_I(x, y) > \Upsilon$  then
7:          $R_{pay}(x, y)= M_I(x, y)-\lambda_c$ 
8:       else
9:          $R_{pay}(x, y)=-\lambda_C$ 
10:      end if
11:    end for
12:  end for
13:  return  $R_{pay}(x, y)$ 
14: end procedure

```

---

Fonte: o autor, (2016).

Uma vez que  $R_{pay}(x, y)$  foi determinada a função  $V$  para todos  $x, y \in M_{mapa}$  é dado por:

$$V_k(x, y) = \gamma \max_{\delta} \left[ R(x, y) + \kappa \right] \quad (4.5)$$

$$\kappa = \sum_{x', y'} V_{k-1}(x', y') \mathbf{P}(x', y' | \delta, x, y) - \lambda_M(\delta) \quad (4.6)$$

onde  $x' \in [1, 2, \dots, x_{max}]$  e  $y' \in [1, 2, \dots, y_{max}]$ , e  $\lambda_M$  e o custo da direção do movimento, e como essa função é calculada recursivamente até  $V_k(x, y) = V_{k-1}(x, y)$ ,  $\gamma < 1$  é uma constante para garantir a convergência. A probabilidade de que a ação  $\delta$  é realmente executada é representada por  $\mathbf{P}(x', y' | \delta, x, y)$ . Mas como para este caso é considerado uma ação determinística  $\mathbf{P}(x', y' | \delta, x, y) = 1$ , assim a equação (4.6) é simplificada em :

$$V_k(x, y) = \gamma \max_{\delta} \left[ R(x, y) + \sum_{x', y'} V_{k-1}(x', y') - \lambda_M \right] \quad (4.7)$$

Os movimentos que o veículo pode realizar é determinado por  $\pi_{pol} : (x, y) \rightarrow \delta$  é assumido como: norte (N), sul (S), leste (E) e oeste (W) e as diagonais nordeste (NE),



sudeste (SE), sudoeste (SW) e noroeste (NW). Essas movimentos são representadas pelos seguintes valores:

$$\begin{aligned}
 \delta_N &= \begin{bmatrix} 0 & 1 \end{bmatrix} & \uparrow \\
 \delta_S &= \begin{bmatrix} 0 & -1 \end{bmatrix} & \downarrow \\
 \delta_W &= \begin{bmatrix} -1 & 0 \end{bmatrix} & \leftarrow \\
 \delta_E &= \begin{bmatrix} 1 & 0 \end{bmatrix} & \rightarrow \\
 \delta_{NW} &= \begin{bmatrix} -1 & 1 \end{bmatrix} & \swarrow \\
 \delta_{NE} &= \begin{bmatrix} 1 & 1 \end{bmatrix} & \nearrow \\
 \delta_{SE} &= \begin{bmatrix} 1 & -1 \end{bmatrix} & \searrow \\
 \delta_{SW} &= \begin{bmatrix} -1 & -1 \end{bmatrix} & \swarrow
 \end{aligned} \tag{4.8}$$

e definem o custo de movimento como  $\lambda_M(\delta) = \|\delta\|_2$ .

Assim uma vez que  $V$  tenha convergido, as melhores *polícies* são computadas para todo o mapa, ou para apenas uma posição inicial por:

$$\delta(x, y) = \arg \max_{\delta \in \pi_{pol}} [V(x, y)] \tag{4.9}$$

## 4.6 Resultados

Nesta seção são apresentados todos os resultados obtidos utilizando a técnica de localização baseada no Filtro de Partículas, os quais foram obtidos através de simulações realizadas no *software MATLAB*. Durante as simulações o veículo aquático foi mantido em uma profundidade de  $d = 20$  m, e as partículas foram inicializadas ao redor da posição, supostamente, inicial do veículo que foi obtida através de um *GPS* antes da submersão. O alcance máximo do sonar foi limitado em  $d_{max} = 25$  m. Foram utilizadas duas trajetórias, a primeira com posição inicial em  $(10, 7)$  e posição final em  $(20, 250)$ . Para a segunda trajetória utilizou-se como condições inicial  $(10, 90)$ , e final  $(20, 250)$ . Os valores de média  $\mu$  e covariância  $\sigma$  utilizados para simular o ruído branco dos sensores são apresentados na tabela 4. Para o Filtro de Partículas foram utilizados os parâmetros descritos na tabela 5.

Tabela 4 – Tabela com os valores de media ( $\mu$ ) e covariância ( $\sigma$ ) utilizados nos ruídos dos sensores

Ruído	Valores ( $\mathcal{N}(\mu, \sigma)$ )
$\varepsilon_{acc}$ (acelerômetro)	$\mathcal{N}(0.01, 0.001)$
$\varepsilon_{gy}$ (giroscópio)	$\mathcal{N}(0.01, 0.1)$
$\varepsilon_z$ (sonar)	$\mathcal{N}(0.01, 0.2)$
$\varepsilon_r$ (boia)	$\mathcal{N}(0.01, 0.1)$

Fonte: o autor (2016).

Tabela 5 – Tabela com os parâmetros utilizado no Filtro de Partículas

Parâmetro	Valores
$\alpha_1$	2
$\alpha_2$	2
$\alpha_3$	20
$\alpha_4$	2
M (número de partículas)	200
<i>ResampleTime</i> (tempo de correção)	a cada 130 amostras
$\sigma_w$ (covariança da estimação)	2.5

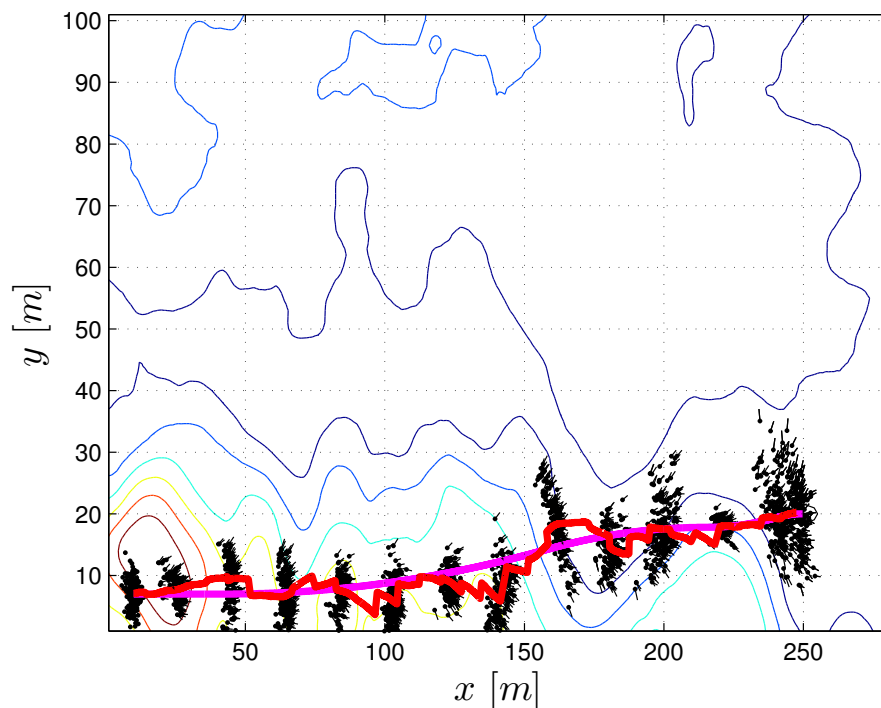
Fonte: o autor (2016).

Nas figuras apresentadas nesta seção a linha roxa representa a trajetória realizada pelo veículo aquático, os pontos na cor preta as partículas do filtro, e as linhas coloridas as diferentes profundidades do mapa. Para todos os resultados foi utilizado o mapa real apresentado na Figura 12B.

#### 4.6.1 Localização Baseada em Terreno (LBT)

O método de Localização Baseado em Terreno utilizado foi o descrito na seção 4.3, o qual utiliza o *IMU* (acelerômetros e giroscópio) para realizar a predição, e o sonar em conjunto com o mapa para fazer a correção. Assim na Figura 53 é apresentado a visão de topo do algoritmo LBT aplicado no mapa real, as partículas são inicializadas ao redor da suposta posição inicial do veículo. Conforme o veículo movimentar-se as partículas começam a se dispersar pelo mapa, para poder lidar com as incertezas oriundas do IMU o filtro utiliza a medida do sonar real e a profundidade do mapa para poder eliminar as partículas que estão estimando a posição errada. Dessa forma como essa trajetória possui regiões com boa variação de profundidade o filtro consegue convergir para a localização correta do veículo.

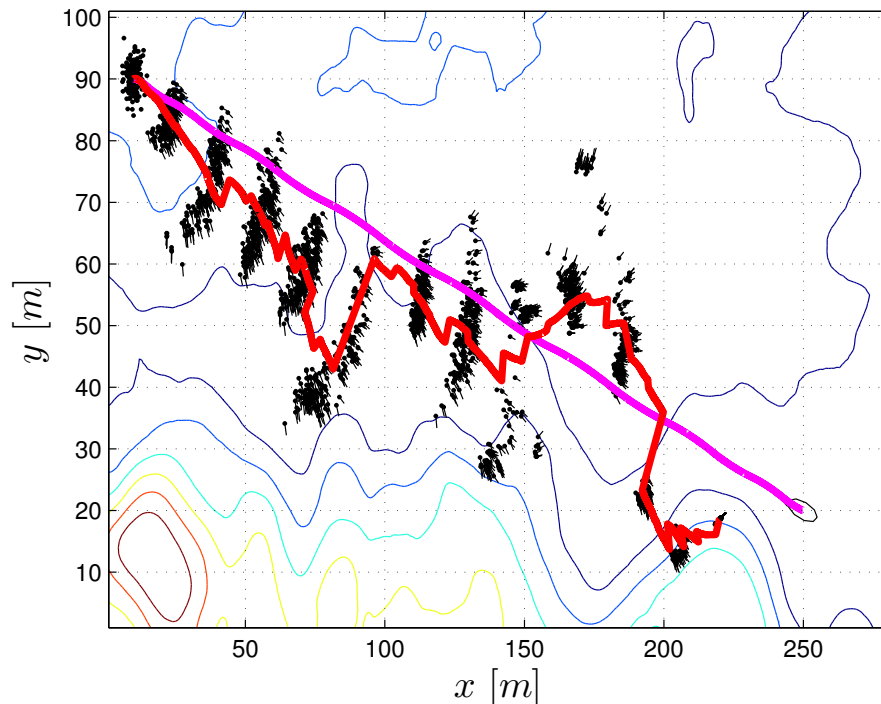
Figura 53 – Resultado obtido para a primeira trajetória com a técnica de LBT, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas



Fonte: o autor, (2016)

Na Figura 54 mostra-se o resultado obtido para a segunda trajetória que é um caminho mais crítico para esse método, devido que o veículo navega por regiões com pouca variação de profundidade. As partículas são inicializadas ao redor da suposta condição inicial do veículo, e se dispersam conforme a movimentação do veículo. Com as diferenças de profundidade do mapa o filtro corrige a estimativa feita pelas partículas, mas a medida que o veículo entra na região com pouca variação de terreno. A técnica já não consegue diferenciar as partículas boas das ruins, com isso as partículas continuam se dispersando indefinidamente ocasionando na má estimativa da posição do veículo.

Figura 54 – Resultado obtido para a segunda trajetória com a técnica de LBT, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas



Fonte: o autor, (2016)

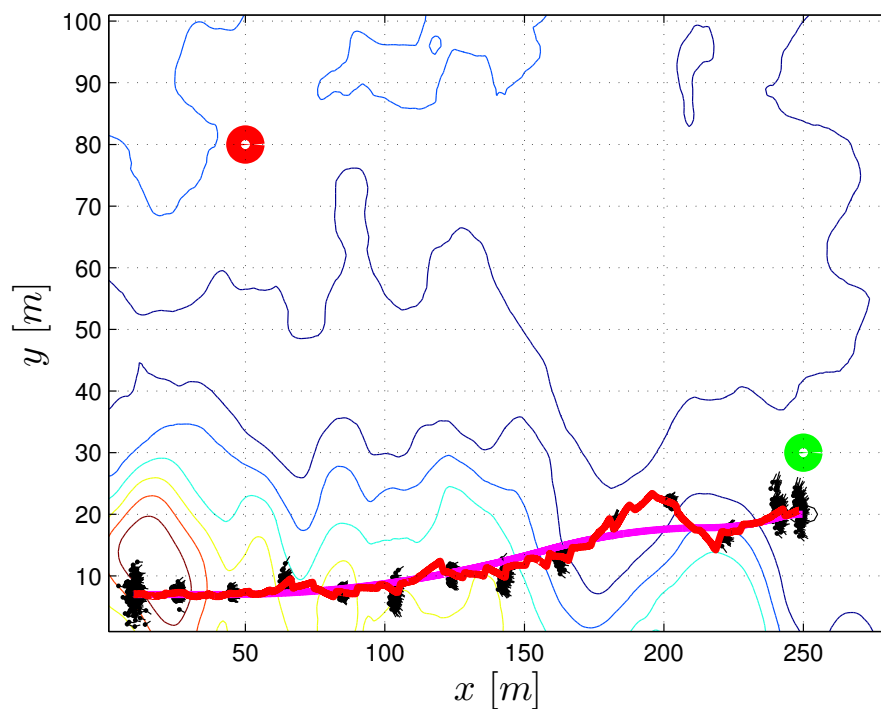
#### 4.6.2 Localização Utilizando Boias

A técnica de localização utilizando boias utiliza uma *IMU* para realizar a predição da posição, e as boias para realizar a correção da estimativa. Foram realizadas simulações utilizando duas e três boias que possuem sua localização fixa e conhecida, e nas figuras estão representadas por círculos coloridos. A primeira boia (cor vermelha) está localizada na posição (50, 80), a segunda (cor azul) está posicionada no centro do mapa em (125, 50), e a terceira (cor verde) está disposta na posição (250, 30). Neste método o mapa não é

considerado para realizar a estimação da localização, é apenas utilizado para determinar a localização das boias.

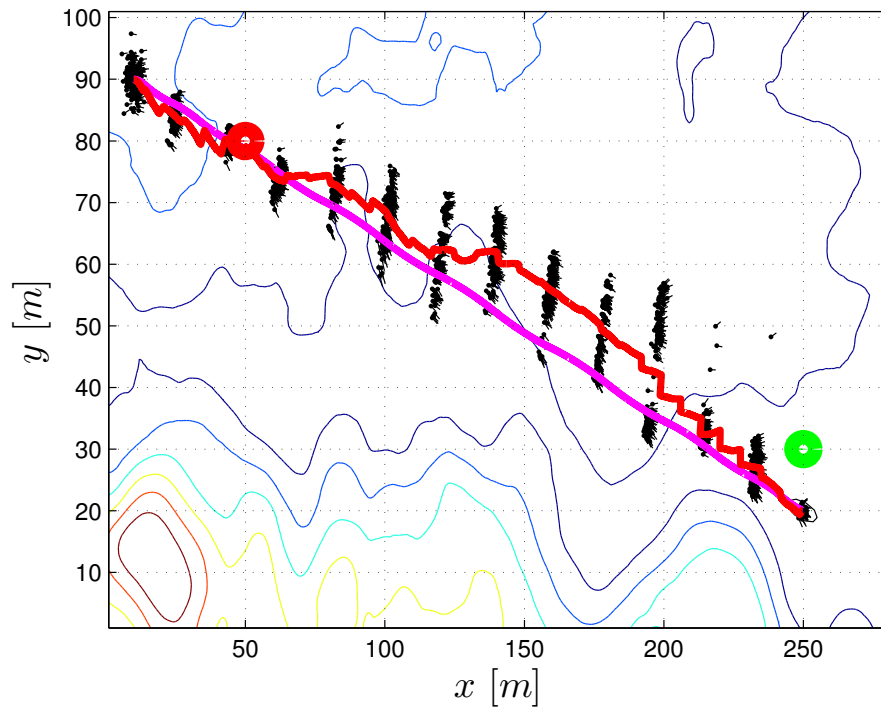
Nas Figuras 55 e 56 respectivamente são apresentados os resultados obtidos para a primeira, e segunda trajetória utilizando duas boias. O filtro começa a estimação posicionando uma nuvem de partículas de forma aleatória ao redor da suposta posição inicial do veículo. Logo, as partículas começam a se dispersar conforme a movimentação do veículo, assim para corrigir a imprecisão da *IMU*, essa técnica utiliza os dados oriundos das boias. Entretanto, pode-se notar que a medida que o veículo se distancia da primeira boia as partículas perdem-se, mas logo que a distancia da segunda boia diminui o filtro consegue acertar novamente a estimação.

Figura 55 – Resultado obtido para a primeira trajetória utilizando o Filtro de Partículas com duas boias, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas



Fonte: o autor, (2016)

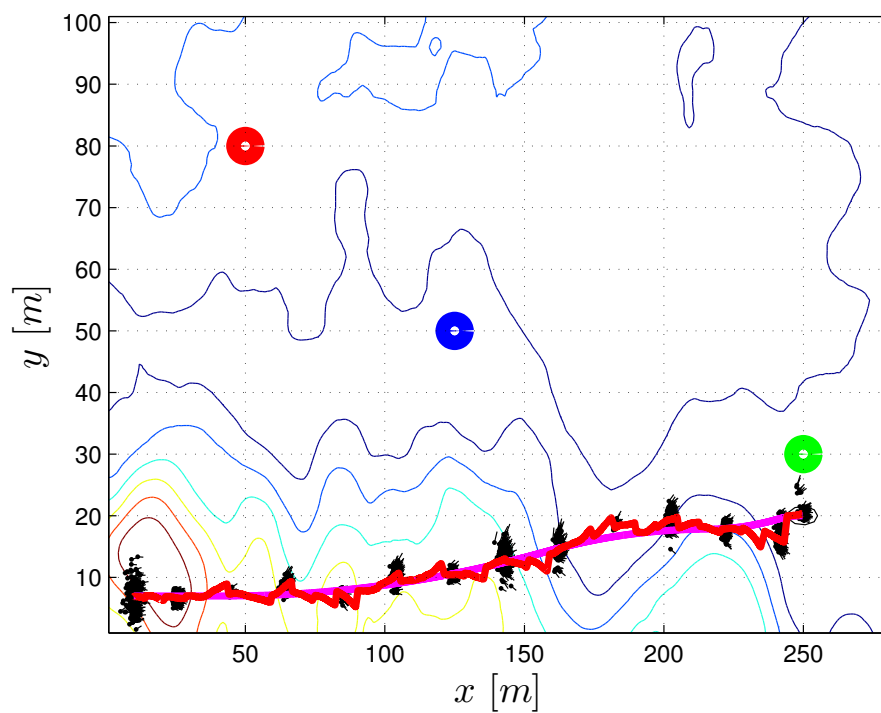
Figura 56 – Resultado obtido para a segunda trajetória utilizando o Filtro de Partículas com duas boias, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas



Fonte: o autor, (2016)

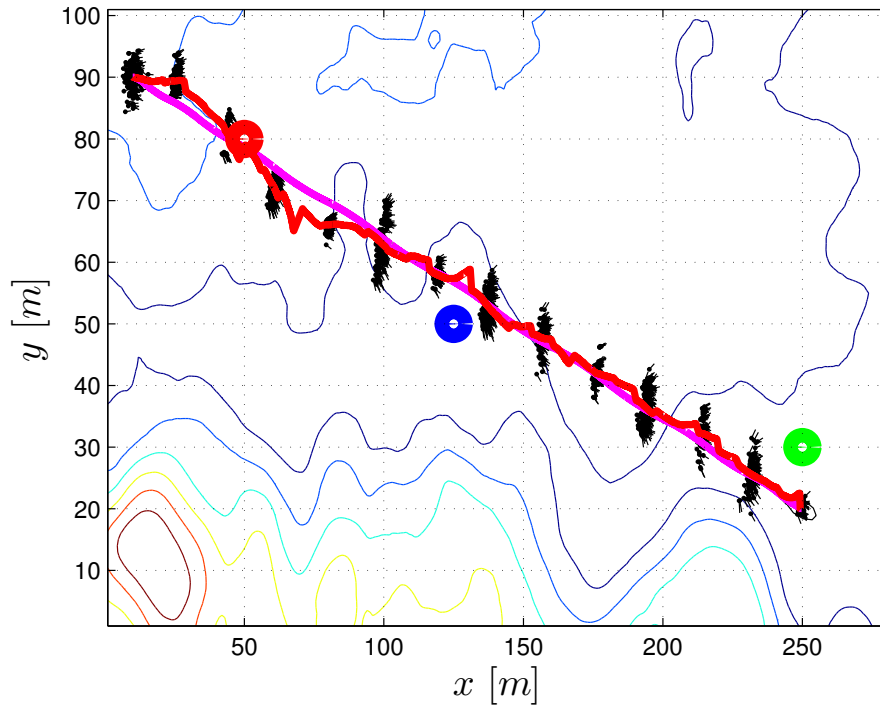
Os resultados obtidos utilizando três boias para a primeira, e segunda trajetória são respectivamente apresentados nas Figuras 57 e 58. Pode-se notar que nesta abordagem não ocorre o problemas das partículas se perderem a medida que se distanciam das boias, sendo assim obteve-se um resultado mais robusto em relação a utilizar duas boias.

Figura 57 – Resultado obtido para a primeira trajetória utilizando o Filtro de Partículas com três boias, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas



Fonte: o autor, (2016)

Figura 58 – Resultado obtido para a segunda trajetória utilizando o Filtro de Partículas com três boias, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas



Fonte: o autor, (2016)

### 4.6.3 LBT em Conjunto com Boias

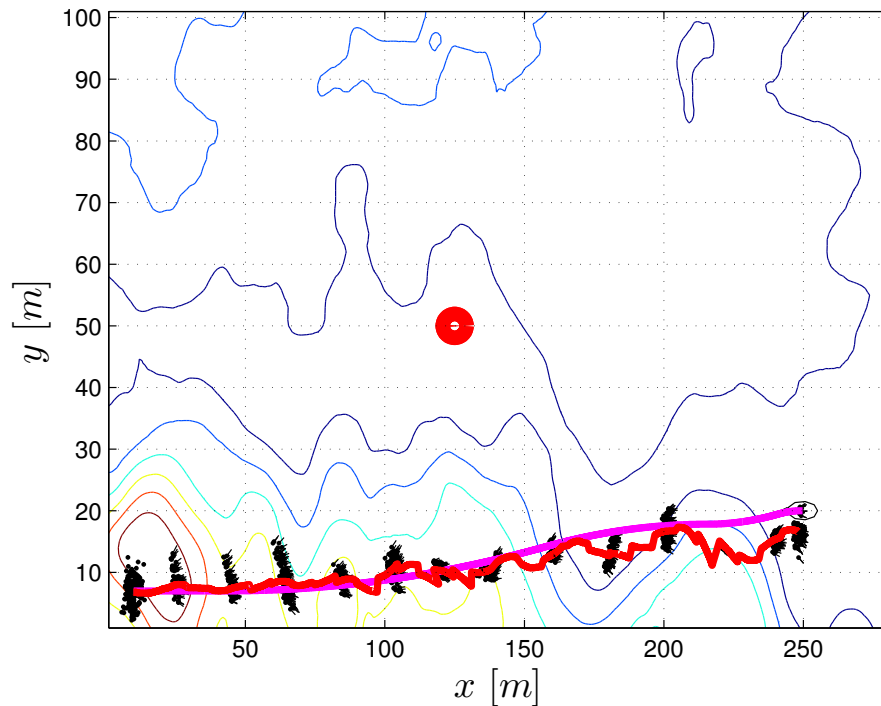
Para solucionar o problema da técnica de LBT optou-se por adicionar uma boia na etapa de correção do filtro. Desse modo, no momento em que o filtro realiza a correção da estimação são calculado dois pesos  $w_k^M$ , um referente a medição do sonar e a profundidade do mapa  $\omega_{sonar}^M$ , e outro para a medição da boia  $\omega_r^M$ . Assim o peso final  $w_k^M$  atribuído para cada partícula é calculado através da seguinte maneira:

$$w_k^M = \omega_{sonar}^M \omega_r^M. \quad (4.10)$$

Com essa abordagem foi possível adicionar mais informação para o filtro no passo de correção, e com isso aumentar a qualidade da estimação e também diminuir a sua dependência da variação do terreno. Os resultados obtidos utilizando esse método para a primeira trajetória são apresentados na Figura 59, e pode-se ver que as partícula foram capazes de estimar a localização do veículo. Essa trajetória não é crítica para esse método devido que o veículo passa por regiões com variação de profundidade.



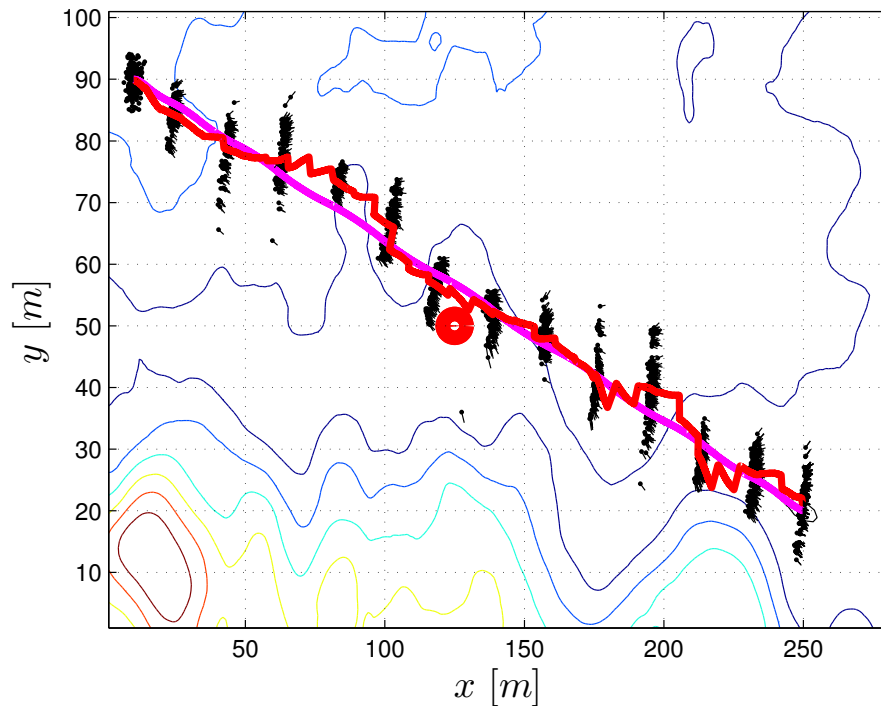
Figura 59 – Resultado obtido para a primeira trajetória com a técnica de LBT em conjunto com o uma boia, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas



Fonte: o autor, (2016)

Na Figura 60 mostra-se o resultado adquirido para a segunda trajetória, que é a mais crítica para o filtro, devido a pouca variação de terreno. Foi demonstrado que ao utilizar apenas a técnica de LBT nessa trajetória as partículas perdem-se e não conseguem estimar a posição correta. Já para o método de LBT com boia esse erro não ocorre, e as partículas conseguem estimar a posição do veículo.

Figura 60 – Resultado obtido para a segunda trajetória com a técnica de LBT em conjunto com uma boia, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas



Fonte: o autor, (2016)

#### 4.6.4 LBT em Conjunto com a Técnica de Geração de Trajetória

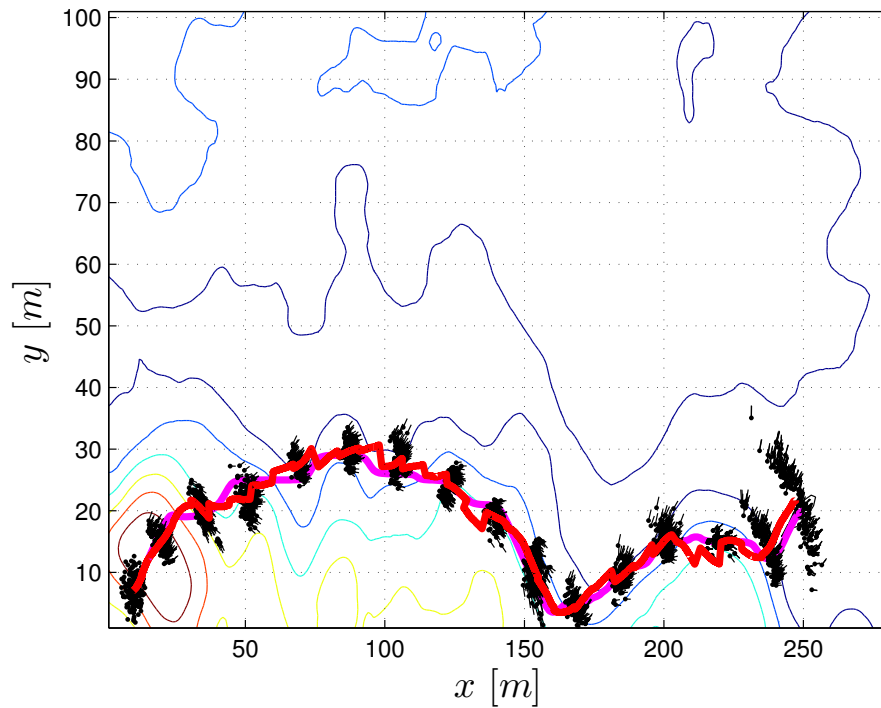
Outra solução que foi desenvolvida para contornar o problema do método LBT foi utilizar um algoritmo de GT para forçar o veículo a se movimentar por regiões do mapa com alta variação de profundidade. Desse modo, antes do veículo começar sua navegação a técnica de GT é acionada e determina qual é a trajetória que o veículo deve percorrer em função de sua posição inicial e final, ou seja o algoritmo de GT determina a melhor rota para o sistema de localização em função do terreno. Com o objetivo de priorizar as regiões do mapa com grande variações de terreno foram utilizados os parâmetros mostrados na tabela 6 para configurar o algoritmo de geração de trajetória.

Tabela 6 – Tabela com os parâmetros utilizado no algoritmo de geração de trajetória. Linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas

Parâmetro	Valores
$\lambda_G$ (custo de chegar na posição final)	100
$\lambda_C$ (custo de chegar em qualquer outro ponto)	1
$\Upsilon$ (limiar de importância do terreno)	0.3
$\gamma$ (fator de desconto)	1

O resultado obtido para a primeira trajetória utilizando o algoritmo de GT em conjunto com o método LBT é apresentado na Figura 61. A linha roxa representa a trajetória calculado pelo GT e também o trajeto realizado pelo veículo, nota-se que o caminho calculado é maior do que o apresentado nos resultados anteriores, mas forçou o veículo a navegar por uma região com maior variação de profundidade. Na Figura 61 é possível notar que o maior número de partículas ficaram posicionadas ao redor da posição real do veículo.

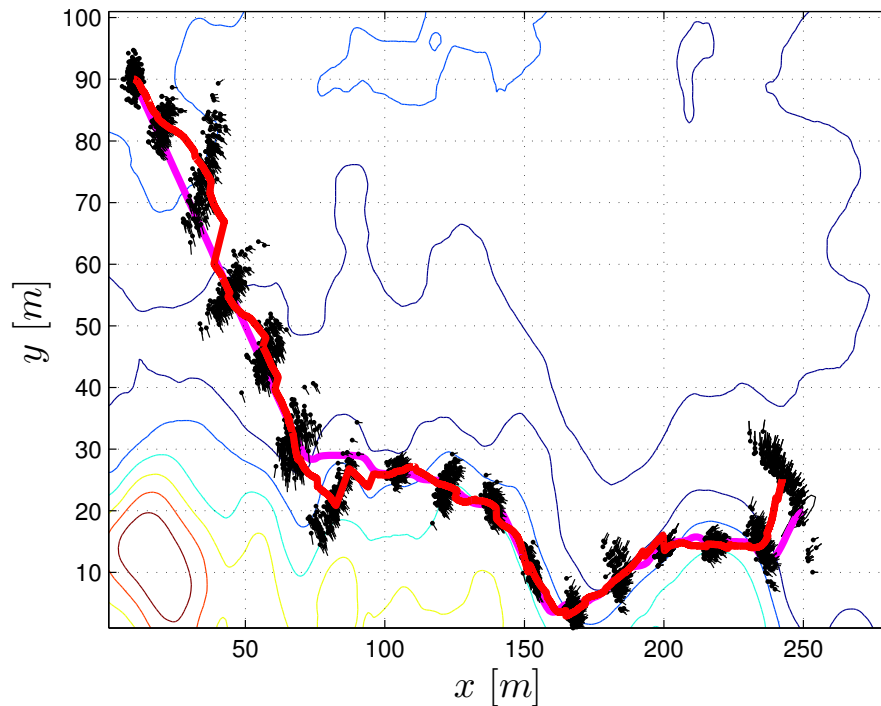
Figura 61 – Resultado obtido para a primeira trajetória com a técnica de LBT em conjunto com o algoritmo de geração de trajetória, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas



Fonte: o autor, (2016)

Para a segunda trajetória os resultados obtidos utilizando a técnica LBT em conjunto com o algoritmo de GT estão representados na Figura 62. O caminho gerado pela técnica de GT forçou o veículo a realizar o caminho mais longo mas com a maior variação de profundidade, fazendo com que as partículas conseguissem convergir para a posição correta.

Figura 62 – Resultado obtido para a segunda trajetória com a técnica de LBT em conjunto com o algoritmo de geração de trajetória, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas

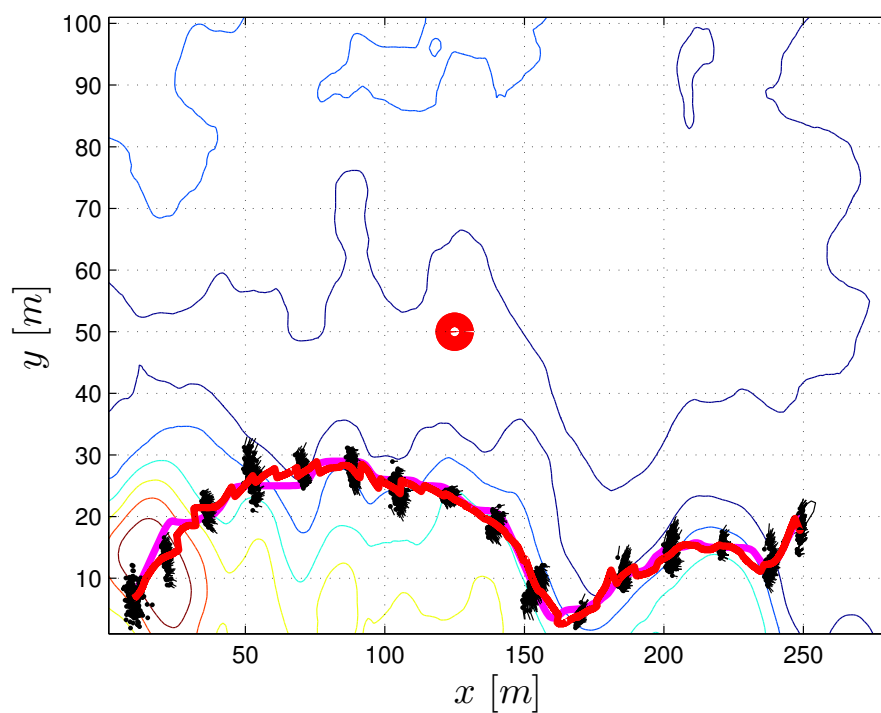


Fonte: o autor, (2016)

#### 4.6.5 LBT em conjunto com a Técnica de Geração de Trajetória e com Boias

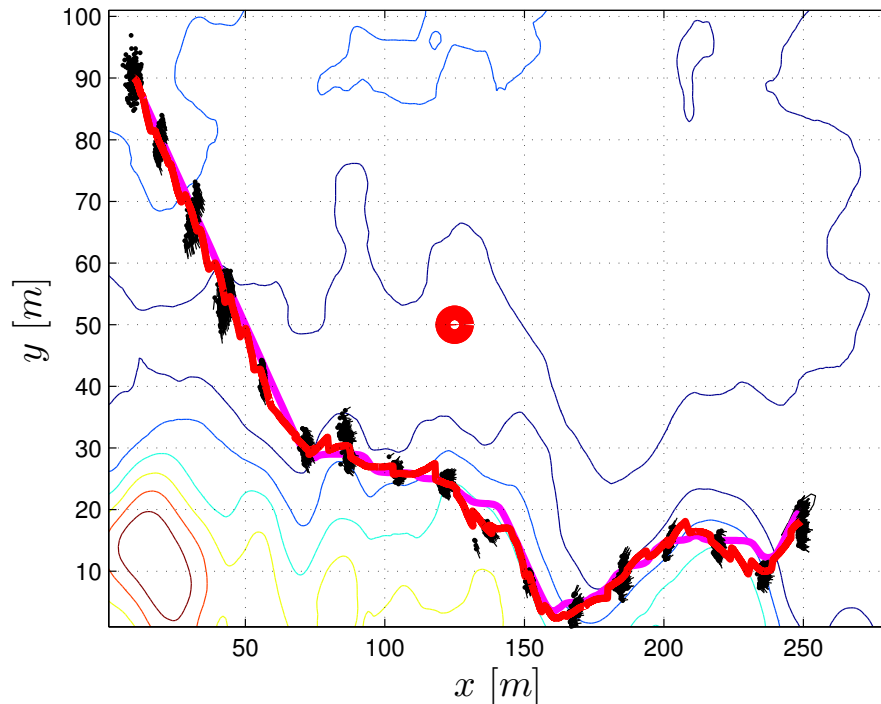
A última abordagem desenvolvida utilizando o Filtro de Partículas foi realizar a combinação do método de LBT com o algoritmo de geração de trajetória, e também uma boia localizada no centro do mapa na posição (125, 50). Desse modo além do veículo navegar por uma região com grande variação de terreno o filtro possui mais um sensor para obter informação e realizar o passo de correção. Os resultados obtidos para a primeira e segunda trajetória respectivamente são apresentados nas Figuras 63 e 64. Para as duas trajetórias essa abordagem conseguiu estimar a posição do veículo sem permitir que as partículas se extraviassem pelo mapa.

Figura 63 – Resultado obtido para a primeira trajetória com a técnica de LBT em conjunto com o algoritmo de geração de trajetória, e o sensor GIB, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas



Fonte: o autor, (2016)

Figura 64 – Resultado obtido para a segunda trajetória com a técnica de LBT em conjunto com o algoritmo de geração de trajetória, e o sensor GIB, onde a linha roxa representa a trajetória real realizada pelo veículo e a linha vermelha representa a trajetória estimada pelo filtro através da média de todas as partículas



Fonte: o autor, (2016)

## 4.7 Análise de Desempenho

Para analisar o desempenho das técnicas de localização baseadas no Filtro de Partículas foi feito o cálculo do erro médio euclidiano da trajetória estimada. Os erros calculados para cada método e para cada trajetória estão apresentados na tabela 7. Pode-se notar que a técnica de LBT apresentou um erro muito grande acima de 5 m para primeira trajetória e acima de 10 m para segunda. Isso ocorre devido a falta de informação que é enviada para o filtro no passo de correção. Para os outros casos o erro obtido foi menor, onde os dois valores mais baixos são das técnicas do FP com três boias, e da LBT com boia e GT.

Tabela 7 – Tabela com o erro média euclidiano da trajetória estimada utilizando os métodos de localização baseados no Filtro de Partículas

Método	Erro (primeira trajetória)	Erro (segunda trajetória)
LBT	5,1 m	13,7 m
FP com duas boias	1,03 m	5,4 m
FP com três boias	1,3 m	1,6 m
LBT com boias	1,9 m	2,5 m
LBT com GT	3,6 m	1,8 m
LBT com boia e GT	1,3 m	1,9 m

Fonte: o autor (2016)

Foi realizado o cálculo do erro da trajetória euclidiana com a seguinte expressão:

$$e_T = \frac{1}{N} \sum_i^N \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \quad (4.11)$$

onde,  $N$  representa o número de amostras,  $\hat{x}$  e  $\hat{y}$  o estado estimado pelo filtro,  $x$  e  $y$  o valor real oriundo do modelo do veículo.

Foi feita a análise de desempenho da estimação do ângulo  $\psi$ , através do cálculo da média do erro. Os valores obtidos são apresentados na tabela 8, e pode-se notar que os maiores valores de erro encontrado foram para o método LBT. As técnicas do FP com três boias, e LBT com boia e GT foram as que demonstraram o melhor desempenho.

Tabela 8 – Tabela com o erro médio do ângulo  $\psi$  estimado utilizando os métodos de localização baseados no Filtro de Partículas

Método	Erro (primeira trajetória)	Erro (segunda trajetória)
LBT	17,5°	21,5°
FP com duas boias	9,8°	15,7°
FP com três boias	11,9°	13,21°
LBT com boias	12,1°	12,69°
LBT com GT	14,38°	11,48°
LBT com boia e GT	12,01°	12,69°

Fonte: o autor (2016)

O cálculo do erro de estimação do ângulo  $\psi$  foi feito da seguinte maneira:

$$e_\psi = \frac{1}{N} \sum_i^N \sqrt{(\psi_i - \hat{\psi}_i)^2}, \quad (4.12)$$



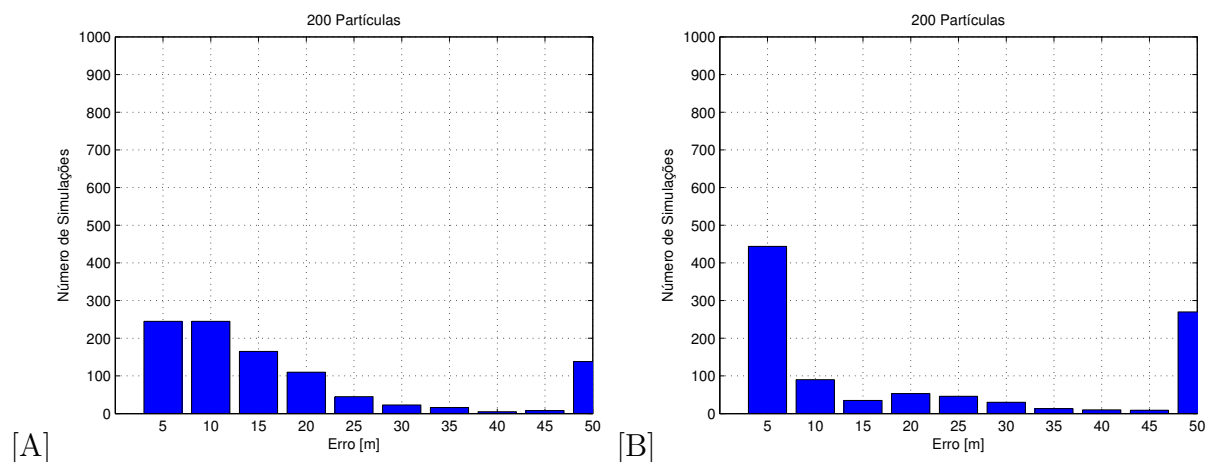
tal que,  $N$  representa o número de amostras,  $\hat{\psi}$  o estado estimado pelo filtro e  $\psi$  o valor real oriundo do modelo do veículo.

## 4.8 Análise de Convergência

Como o Filtro de Partículas depende de números aleatórios para estimar os estados, foram realizadas mil simulações para analisar a convergência da técnica de LBT e também analisar se o algoritmo de GT melhora a estimação do filtro. Para isso foram criados setes grupos com 200, 500, 1000, 2000, 3000, 4000 e 5000 partículas. Desse modo para cada grupo foram realizadas mil simulações e para cada uma calculou-se o erro médio da trajetória estimada segundo a equação 4.11. A segunda trajetória foi utilizada em todas as simulações já que é a mais crítica para o método de localização e também porque era necessário descobrir se o algoritmo de GT apresentou alguma melhora no desempenho do filtro. Assim todos os valores de erro maiores que 50 m foram descartados.

Na Figura 65 são apresentados os resultados obtidos para o grupo de 200 partículas, e pode-se notar que para técnica de LBT sem o algoritmo de GT (Figura 65 A) obteve-se em torno de 200 simulações com erro perto de 5 metros. Com a técnica de LBT em conjunto com o algoritmo de GT o número de simulações com erro perto de 5 metros aumentou para cerca de 400.

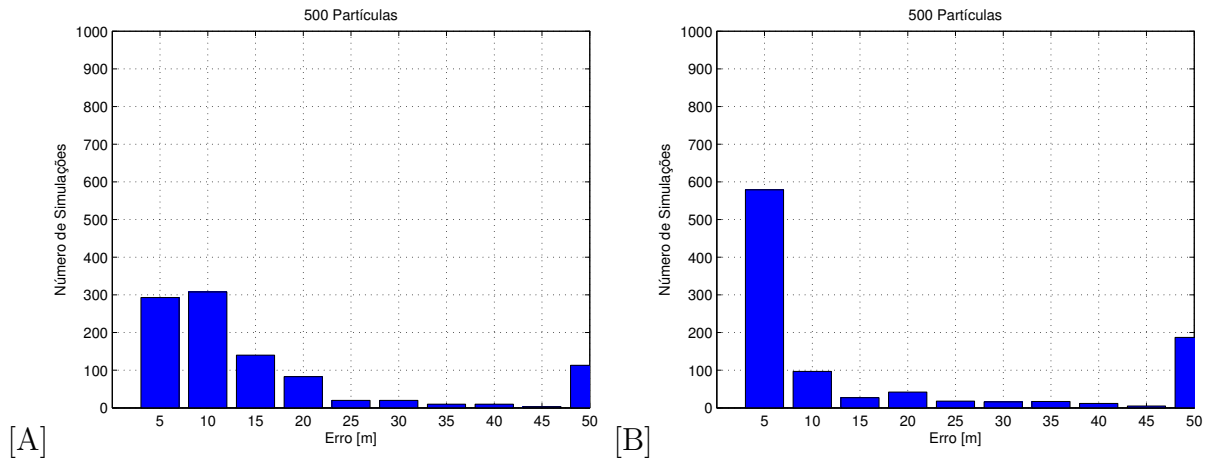
Figura 65 – Resultados obtidos com mil simulações para 200 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.



Fonte: o autor, (2016)

Para o grupo de 500 partículas os resultados são descritos na Figura 66, onde é possível observar que o método de LBT (Figura 66 A) apresentou um pouco menos que 300 simulações com erro próximo a 5 metros. Já a técnica de LBT com o algoritmo de GT (Figura 66 B) obteve mais de 500 simulações com erro próximo a 5m.

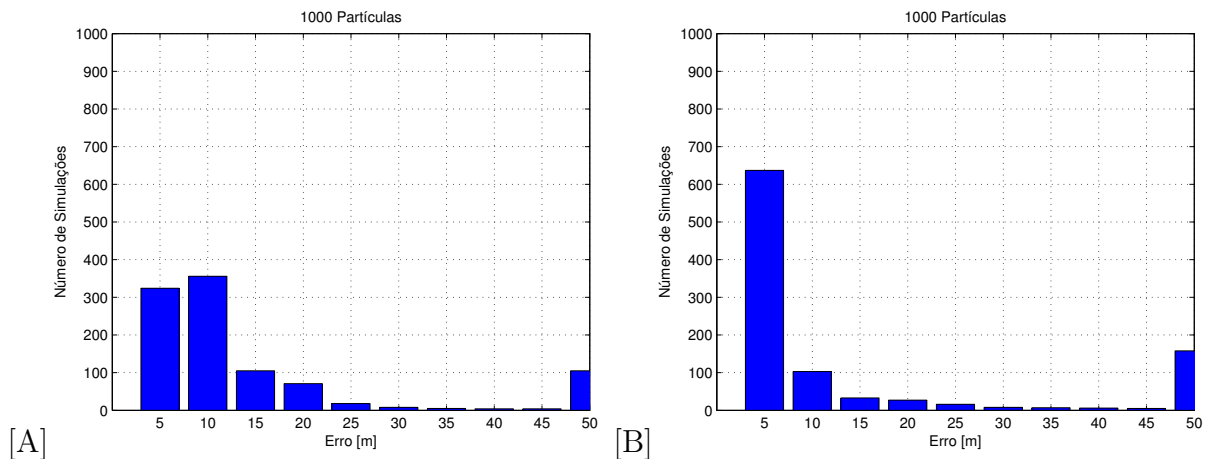
Figura 66 – Resultados obtidos com mil simulações para 500 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.



Fonte: o autor, (2016)

Com 1000 partículas foram obtidos os resultados apresentados na Figura 67, onde a técnica de LBT (Figura 67 A) apresentou um aumento no número de simulações com erro próximo de 5 e 10 metros que ficou em torno de 300 simulações. Mas já o método de LBT em conjunto com o algoritmo de GT (Figura 67 B) obteve um pouco mais de 600 simulações com erro aproximado de 5 metros.

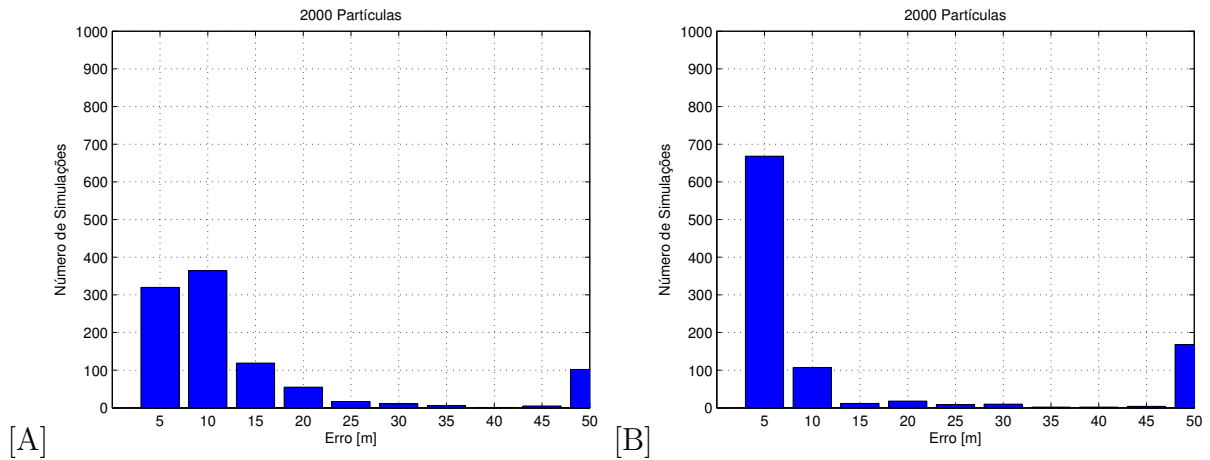
Figura 67 – Resultados obtidos com mil simulações para 1000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.



Fonte: o autor, (2016)

Para o grupo de 2000 partículas os resultados que foram obtidos são demonstrados na Figura 68. Em torno de 300 simulações com erro próximo de 5 e 10 metros foram obtidas para o método de LBT (Figura 68 A), já a técnica de LBT em conjunto com o algoritmo de GT (Figura 68 B) atingiu um número maior de simulações que ficou em torno de 650 com erro aproximado de 5 metros.

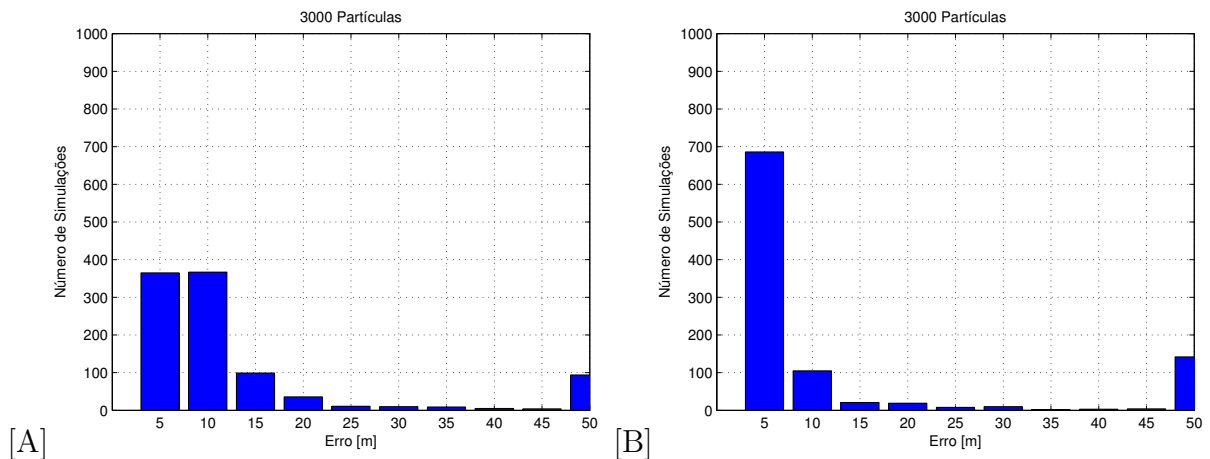
Figura 68 – Resultados obtidos com mil simulações para 2000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.



Fonte: o autor, (2016)

Com 3000 partículas adquiriu-se os resultados demonstrados na Figura 69, onde com o método de LBT (Figura 69 A) em torno de 350 simulações com erro próximo de 5 e 10 metros foram obtidas. Já para o grupo de 3000 que utiliza o método de LBT em conjunto com o algoritmo de GT (Figura 69 B) foi obtido um número próximo de 700 simulações com erro perto de 5 metros.

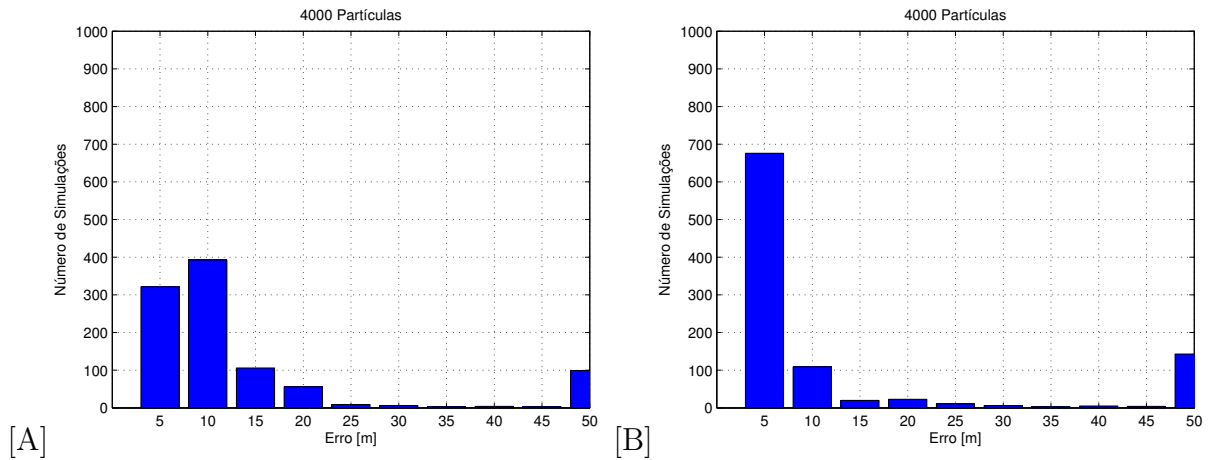
Figura 69 – Resultados obtidos com mil simulações para 3000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.



Fonte: o autor, (2016)

Para o grupo de 4000, os resultados atingidos são demonstrados na Figura 70, assim para o método de LBT em torno de 400 simulações com erro próximo de 10 metros foram obtidas, e também 300 simulações com erro perot de 5 metros. Com a técnica de LBT em conjunto com o algoritmo de GT perto de 700 simulações com erro próximo de 5 metros foram adquiridas.

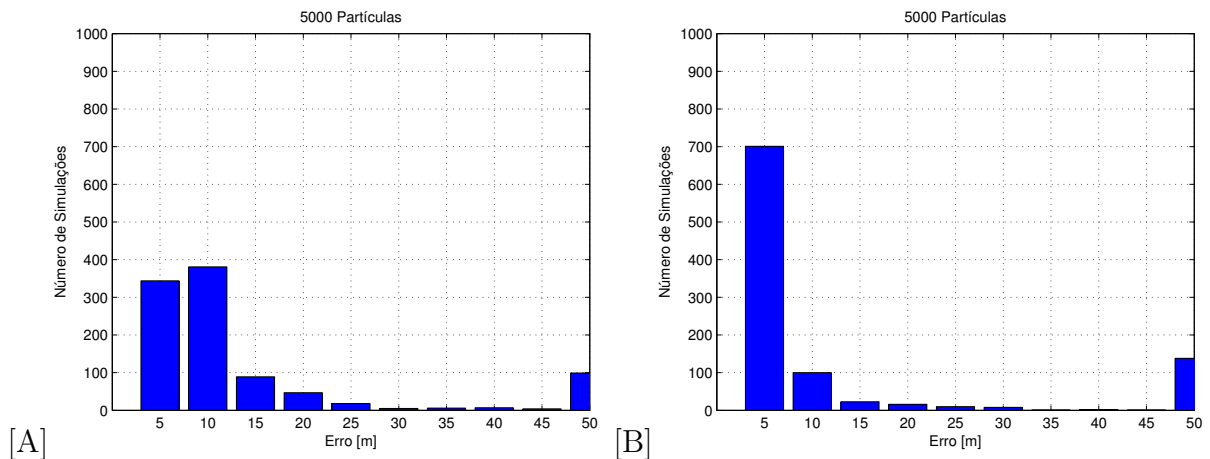
Figura 70 – Resultados obtidos com mil simulações para 4000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.



Fonte: o autor, (2016)

Com 5000 partículas os resultados adquiridos são demonstrados na Figura 71, onde com o método de LBT foram obtidas em torno de 400 simulações com erro próximo de 10 metros, e 350 simulações com erro erro perto de 5 metros. Já para a técnica de LBT em conjunto com o algoritmo de GT perto de 700 simulações com erro próximo de 5 metros foram adquiridas.

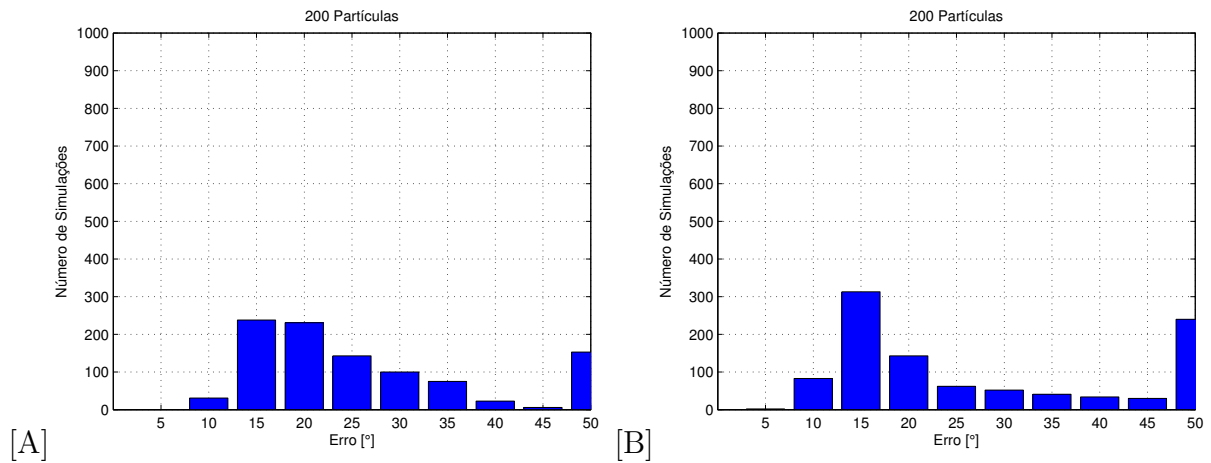
Figura 71 – Resultados obtidos com mil simulações para 5000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.



Fonte: o autor, (2016)

Os resultados obtidos na estimação do ângulo  $\psi$  são demonstrados nas figuras abaixo. Para o método de LBT com 200 partículas (Figura 72 A) foram obtidas em torno de 250 simulações com erro próximo a  $15^\circ$ . Já para a técnica de LBT em conjunto com o algoritmo de GT (Figura 72 B) foi obtido um valor próximo de 300 simulações com erro em torno de  $15^\circ$ .

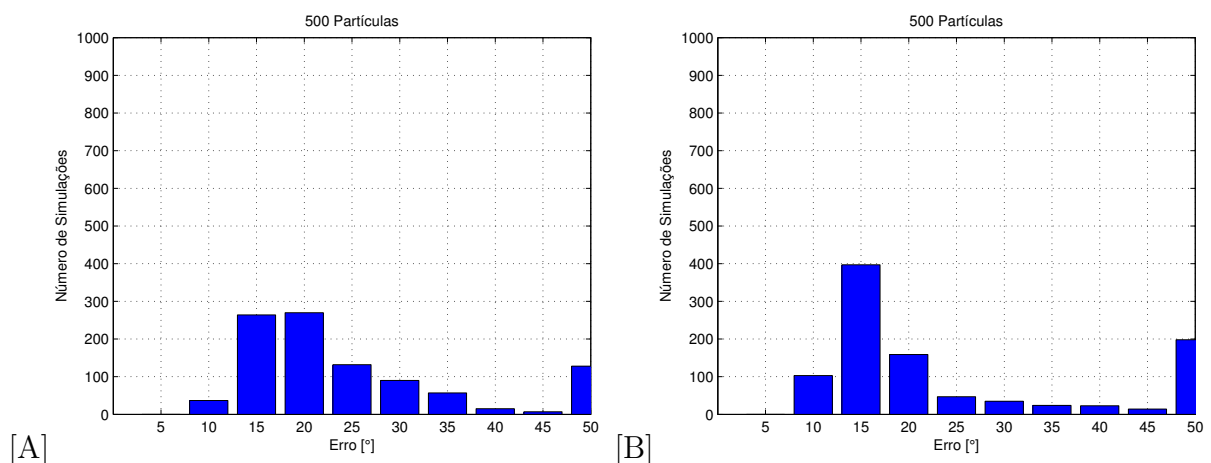
Figura 72 – Resultados obtidos na estimação do ângulo  $\psi$  com mil simulações para 200 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.



Fonte: o autor, (2016)

Para o grupo de 500 partículas os resultados atingidos na estimação do ângulo  $\psi$  são demonstrados na Figura 73, onde obteve-se em torno de 250 simulações com erro próximo de  $15^\circ$  para o método de LBT (Figura 73 A). Com a técnica de LBT em conjunto com o algoritmo de GT foram adquiridas 400 simulações com erro em torno de  $15^\circ$ .

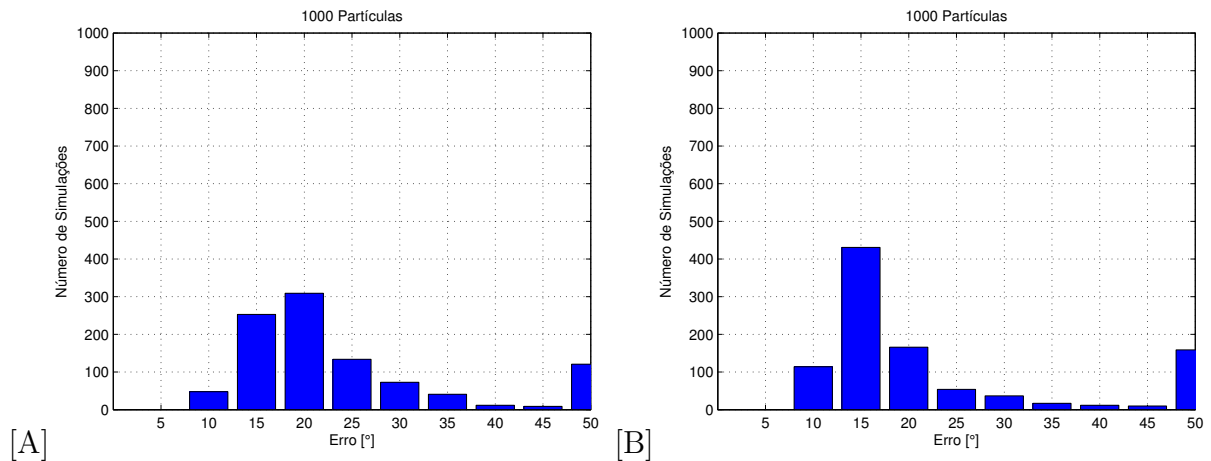
Figura 73 – Resultados obtidos na estimação do ângulo  $\psi$  com mil simulações para 500 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.



Fonte: o autor, (2016)

Com 1000 partículas os resultados obtido são apresentados na Figura 74, assim para o método de LBT (Figura 74 A) em torno de 250 simulações foram adquiridas com erro próximo de  $15^\circ$ . Já para o método de LBT com GT (Figura 74 B) um valor perto de 450 simulações com erro em torno de  $15^\circ$  foi obtido.

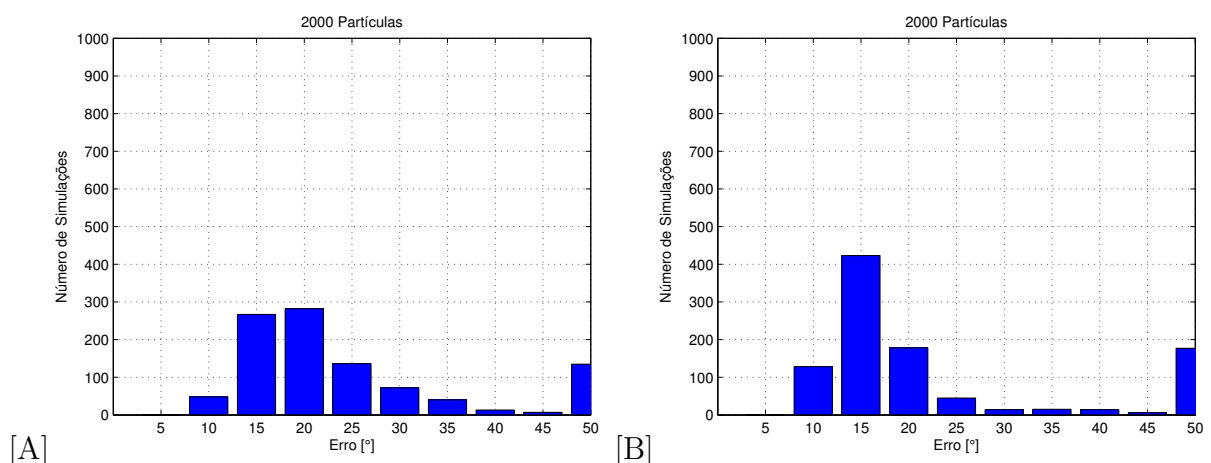
Figura 74 – Resultados obtidos na estimação do ângulo  $\psi$  com mil simulações para 1000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.



Fonte: o autor, (2016)

Para o grupo de 2000 partículas os resultados obtidos são descritos na Figura 75, onde para o método de LBT (Figura 75 A) foram adquiridas em torno de 250 simulações com erro perto de  $15^\circ$ . Com a técnica de LBT em conjunto com o algoritmo de GT (Figura 75 B) um valor próximo de 450 simulações com cerca de  $15^\circ$  de erro foi atingido.

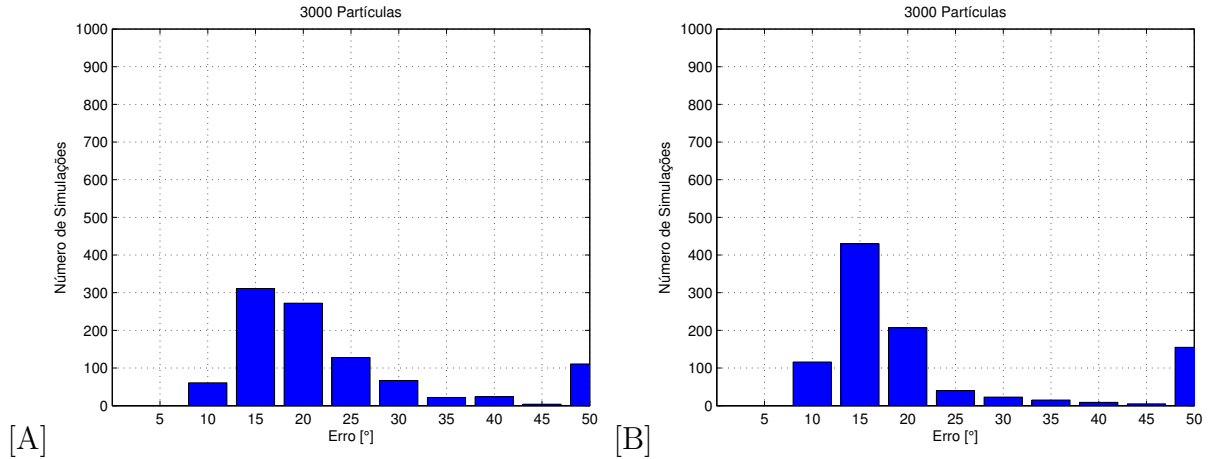
Figura 75 – Resultados obtidos na estimação do ângulo  $\psi$  com mil simulações para 2000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.



Fonte: o autor, (2016)

Com 3000 partículas os resultados adquiridos são apresentados na Figura 76. Para a técnica de LBT (Figura 76 A) um pouco mais de 300 simulações com erro próximo de  $15^\circ$  foram obtidas, já para o método de LBT com GT (Figura 76 B) um valor perto de 450 simulações com cerca de  $15^\circ$  de erro foram atingidas.

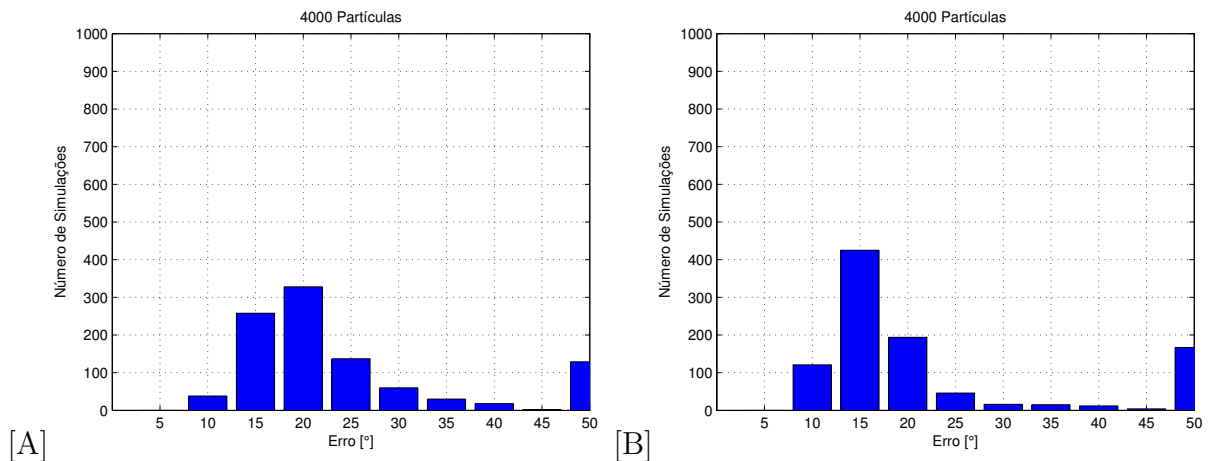
Figura 76 – Resultados obtidos na estimação do ângulo  $\psi$  com mil simulações para 3000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.



Fonte: o autor, (2016)

Para o grupo de 4000 partículas os resultados obtidos são demonstrados na Figura 77, onde para o método de LBT (Figura 77 A) um valor em torno de 250 simulações com erro próximo a  $15^\circ$  foram adquiridas. Já para a técnica de LBT com GT (Figura 77 B) cerca de 450 simulações com erro perto de  $15^\circ$  foram obtidas.

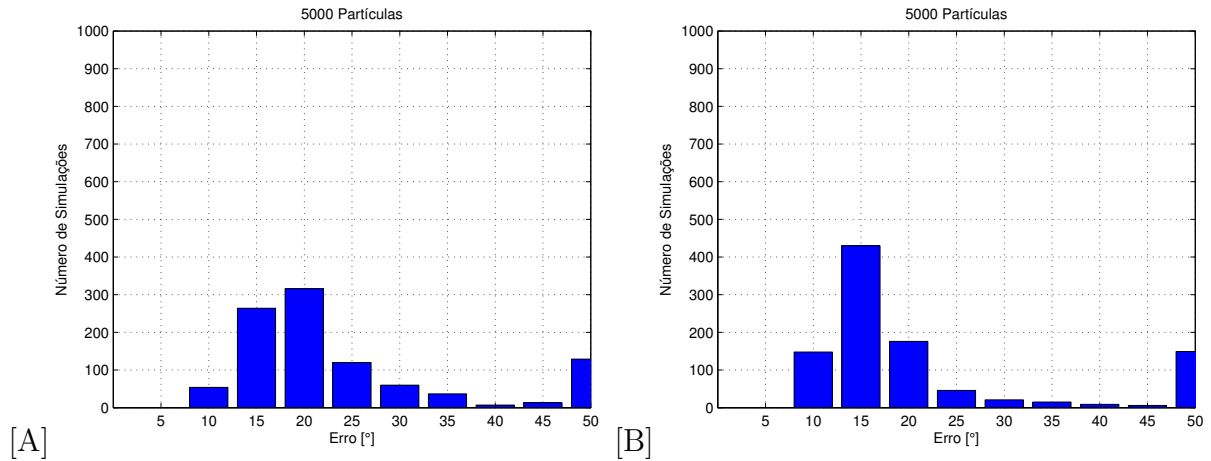
Figura 77 – Resultados obtidos na estimação do ângulo  $\psi$  com mil simulações para 4000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.



Fonte: o autor, (2016)

Os resultados adquiridos para o grupo de 5000 partículas são apresentados na Figura 78, onde para a técnica de LBT um pouco mais de 250 simulações com cerca de  $15^\circ$  de erro foram obtidas. Já para o método de LBT em conjunto com o algoritmo de GT um valor próximo de 450 simulações com erro perto de  $15^\circ$  foram obtidas.

Figura 78 – Resultados obtidos na estimação do ângulo  $\psi$  com mil simulações para 5000 partículas. A) Técnica de LBT. B) Técnica de LBT com o algoritmo de geração de trajetória.



Fonte: o autor, (2016)

## 4.9 Análise de Custo Computacional

O algoritmo do Filtro de Partículas foi implementado em linguagem de programação “C”, para assim calcular o tempo de execução da técnica. A primeira análise foi realizada no *software MATLAB*, em que utilizou-se do pacote *mex-file* para poder simular código em “C”. Assim os modelos matemáticos do veículo e dos sensores foram simulados em linguagem *MATLAB* e seus valores foram utilizados como entrada para o algoritmo do FP. Dessa maneira foi calculado o tempo de execução total da função do FP, através da ferramenta *Run and Time* do *software MATLAB*. Esses tempos de execução foram calculado em um computador com as especificações apresentadas na Tabela xx. Os resultados obtidos são descritos na Tabela 10 e pode-se ver que os tempos de execução utilizando código “C em conjunto com *MATLAB* são bem menores que apenas utilizar *MATLAB*.

Tabela 9 – Especificações do computador utilizado para computar os tempos de execução do FP

<i>Hardware</i>	Especificações
Processador	<i>Intel Core i5-2400 3, 10GHz</i>
Memoria RAM	4GB
Sistema Operacional	<i>Windows 7(64 Bits)</i>



Tabela 10 – Tabela com os tempos de execução do Filtro de Partículas com diferentes quantidades de partículas

Número de partículas	Tempo de execução do código apenas em <i>MATLAB</i> (segundos)	Tempo de execução do código em <i>MATLAB</i> e linguagem “C” (segundos)
200	11,7 s	0,47 s
800	46,8 s	1,7 s
1000	58,6 s	2,1 s
2000	115,5 s	4,3 s
3000	173,2 s	6,5 s
5000	291,2 s	10,8 s

Fonte: o autor (2016)

Outra metodologia utilizada para computar o tempo de execução do filtro foi embarcar o algoritmo desenvolvido em “C” em uma placa micro-controlada. As especificações do processador são as seguintes:

- Microcontrolador : AT91SAM3X8E
- Tensão de operação : 3,3V
- Tensão de entrada recomendada: 7 – 12V
- Tensão de entrada limite : 6 – 16V
- Pinos de *I/O*: 54 ( dos quais 12 fornece saída de *PWM*)
- Pinos de entrada analógicos : 12
- Pinos de saída analógicos : 2 (*DAC*)
- Total DC Output Current on all I/O lines : 130 mA
- Memória flash : 512 KB
- *SRAM* : 96 KB
- *Clock Speed* : 84 MHz

Para determinar o tempo de execução do filtro a cada *loop* do algoritmo um pino do microcontrolador foi setado para 1, e assim com um osciloscópio foi medido o tempo necessário para calcular a estimação. Foram utilizados diferentes valores para o número de partículas, os resultados obtidos são apresentados na Tabela 11. Pode-se notar que o tempo

de execução aumenta conforme aumenta o número de partículas utilizados, dessa maneira com esse algoritmo utilizando o processador especificado acima, e com 5000 partículas a cada 2,8 segundos obtêm-se uma estimativa da *pose* do veículo.

Tabela 11 – Tabela com o tempo de execução de cada *loop* do Filtro de Partículas no processador

Número de partículas	Tempo de execução de um <i>loop</i> (segundos)
200	0,10 s
800	0,40 s
1000	0,48 s
2000	0,95 s
3000	1,4 s
5000	2,8 s

Fonte: o autor (2016)

## 4.10 Comentários Finais

Neste capítulo foram apresentados métodos para solucionar o problema de localizar um veículo aquático utilizando o Filtro de Partículas. A principal técnica desenvolvida foi a Localização Baseada em Terreno, a qual utiliza um mapa batimétrico do local de navegação para compor a estimação da localização. Entretanto, foi apresentado que esse método possui uma deficiência que ocorre quando o veículo transita por regiões do mapa com pouca variação de terreno.

Para solucionar esse problema duas técnicas foram propostas, uma solução em *software* para evitar o emprego de sensores externos, e assim diminuir o custo de implementação, e a outra solução emprega uma boia como sensor externo para ajudar no passo de correção. A solução em *software* desenvolvida utiliza um algoritmo de geração de trajetória que faz com que o veículo navegue por regiões com grande variações de terreno. Para validar essas técnicas propostas simulações foram realizadas utilizando o software *MATLAB*. Também foram realizadas combinações entre os dois métodos para analisar qual possui o melhor desempenho. Todos os resultados apresentados neste capítulo são validos apenas para o mapa utilizado, devido que para um novo mapa é necessário ajustar os parâmetros do filtro e também do algoritmo de GT.

Como o FP depende de números aleatórios para realizar a estimação dos estados foram criados sete grupos com 200, 500, 1000, 2000, 3000, 4000 e 5000 partículas e mil simulações para cada um foram realizadas. Isso foi feito com o objetivo de analisar a

capacidade de convergência da técnica LBT e também verificar se o algoritmo de GT realmente melhora o desempenho do filtro, assim as simulações foram realizadas apenas para a segunda trajetória, já que possui menos variação de terreno (sendo assim a mais crítica para o filtro). Para a estimação da posição do veículo essa metodologia mostrou que o método de LBT não possui uma alta porcentagem de convergência, mas ao adicionar o algoritmo de GT o número de simulações com erro tolerável aumentou. Com isso pode-se concluir que a técnica de GT melhora o desempenho do filtro para trajetória com pouca variação de terreno. Já para estimação do ângulo  $\psi$  nenhum dos métodos apresentou uma boa convergência, isso está associado a falta de informação sobre a orientação do veículo no passo de correção do filtro.

Por último foi realizada uma análise do tempo de execução do algoritmo do FP utilizando o software *MATLAB* em conjunto com linguagem “C”, e também utilizando um microprocessador. Pode-se concluir que dependendo da dinâmica do veículo é possível utilizar esse algoritmo em conjunto com o processador especificado para determinar a localização de um veículo aquático na prática.

## 5 Conclusões e Perspectivas

Este trabalho apresentou técnicas para determinar a posição e orientação de um veículo aquático, onde foram divididas em duas classes. Baseadas no Filtro de Kalman Estendido e baseadas no Filtro de Partículas. No capítulo 3 foi apresentado os métodos desenvolvidos que utilizam o FKE e foram realizadas simulações que mostram que o número recomendado de boias a ser utilizado para obter uma boa estimativa são três. Para melhorar o desempenho do filtro foi desenvolvido um sensor virtual, assim o FKE com boias em conjunto com o sensor virtual apresentou o melhor desempenho com pequenos erros médios de estimativa.

O método de LBT desenvolvido apresentado no capítulo 4 apresentou um problema, que ocorre quando o veículo passa por regiões do mapa com pouca variação de terreno. Para solucionar essa deficiência um método de GT foi desenvolvido que força o veículo a navegar por regiões do mapa com grande variação de profundidade, fazendo o filtro convergir para estimativa correta. Entre as técnicas que utilizam o FP a LBT com GT, LBT com GT e boias, LBT com boias e FP com três boias foram as que apresentaram os melhores desempenhos. Foi realizada uma análise computacional do método LBT, onde o algoritmo do filtro foi desenvolvido em linguagem de programação “C”, e foi embarcado em uma placa micro-controlada. Esse teste mostrou que é possível utilizar esse algoritmo em uma aplicação real utilizando um número alto de partículas.

Como o FP depende de números aleatórios para realizar a estimativa uma análise de convergência para as técnicas de LBT e LBT com GT foi realizada apenas para segunda trajetória, já que é a mais crítica para o filtro. Para isso foram utilizados sete grupos com 200, 500, 1000, 2000, 3000, 4000 e 5000 partículas e mil simulações para cada grupo foi feita. Desse modo para determinar o desempenho das técnicas em cada simulação o erro médio da estimativa foi calculado e seu histograma foi utilizado para realizar a análise de convergência das técnicas. Essa metodologia demonstrou que o método de LBT não possui uma alta porcentagem de convergência para segunda trajetória (a que possui menor variação de terreno). A porcentagem de convergência com erro tolerável aumentou com a adição do algoritmo de GT, assim mostra-se que a Técnica de GT melhora o desempenho do filtro.

Foram desenvolvidas técnicas de localização para um veículo aquático utilizando o FKE e o FP que não dependem do modelo matemático do sistema. As técnicas baseadas no FKE apresentaram melhor desempenho que as baseadas no FP, devido ao pequeno (em torno de centímetros) erro de estimativa obtido. Mas os métodos desenvolvidos que utilizam o FKE demandam de um custo de implementação maior que as técnicas baseadas

no FP, isso porque necessitam no mínimo de três boias para realizar a estimação da localização. Além do baixo custo de implementação as técnicas implementadas com o FP apresentaram uma facilidade no momento de adicionar novos sensores a estimação. Como foi demonstrado neste trabalho basta acrescentar mais uma equação no momento de determinação dos pesos da partículas. O FP exige um desempenho computacional alto quando comparado com o FKE, mas isso hoje em dia não é mais um problema, devido ao avanço da eletrônica. Foi comprovado neste trabalho que é possível implementar um algoritmo utilizando o FP em um processador de baixo custo.

Desse modo pode-se concluir que entre todas as técnicas apresentadas neste trabalho a que melhor se adapta para solucionar o problema de localizar um veículo aquático depende da aplicação final de dos recursos disponíveis. Por exemplo a técnica de LBT com GT é a que possui o melhor desempenho com menor custo, mas fica limitada a certas regiões de navegação. Já a LBT com boia e o FKE com boia são métodos com um custo mais alto mas, que possuem uma liberdade de navegar por qualquer região com restrição de existir boias por perto.

## 5.1 Perspectivas para Trabalhos Futuros

Como perspectivas para trabalhos futuros pode-se citar os seguintes itens:

- Expandir as técnicas de localização para um modelo 3D;
- Encontrar uma alternativa ao utilizar a média das partículas para determinar a estimação final do FP;
- Aumentar os estados do filtro e estimar os distúrbios que afetam o veículo através de um processo probabilístico *Random Walk* como é apresentado por MARTINEZ et al., 2015;
- Desenvolver um método para identificar os *clusters* criados pelas partículas;
- Desenvolver o algoritmo de geração de trajetórias que considera ações probabilísticas do veículo e que seja em tempo real;
- Desenvolver um controlador repetitivo ressonante para seguimento de trajetória e rejeição de distúrbios produzidos por correntes marítimas, já que são considerados periódicos segundo DO; PAN, 2009. Utilizar os estados estimados pelo filtro para fechar a malha de controle;

## Referências

- ALCOCER, A.; OLIVEIRA, P.; PASCOAL, A. Underwater acoustic positioning systems based on buoys with gps. In: *Proceedings of the Eighth European Conference on Underwater Acoustics*. [S.l.: s.n.], 2006. v. 8, p. 1–8. Citado 2 vezes nas páginas 27 e 28.
- AMAZÔNIA Azul. 2016. <<https://www.marinha.mil.br/sic/amazonia-azul.html>>. Acessado: 2016-02-03. Citado na página 21.
- ANONSEN, K. B.; HALLINGSTAD, O. Terrain aided underwater navigation using point mass and particle filters. In: *Proceedings of the IEEE/ION Position Location and Navigation Symposium*. [S.l.: s.n.], 2006. Citado na página 29.
- ARULAMPALAM, M. S. et al. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, IEEE, v. 50, n. 2, p. 174–188, 2002. Citado na página 39.
- CAITI, A. et al. Localization of autonomous underwater vehicles by floating acoustic buoys: a set-membership approach. *Journal of Oceanic Engineering, IEEE*, IEEE, v. 30, n. 1, p. 140–152, 2005. Citado na página 27.
- CARRERAS, M. et al. Vision-based localization of an underwater robot in a structured environment. In: IEEE. *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*. [S.l.], 2003. v. 1, p. 971–976. Citado na página 26.
- DO, K. D.; PAN, J. *Control of ships and underwater vehicles: design for underactuated and nonlinear marine systems*. [S.l.]: Springer Science & Business Media, 2009. Citado na página 108.
- DONOVAN, G. T. Position error correction for an autonomous underwater vehicle inertial navigation system (ins) using a particle filter. *Oceanic Engineering, IEEE Journal of*, IEEE, v. 37, n. 3, p. 431–445, 2012. Citado na página 29.
- EROL, M.; VIEIRA, F.; GERLA, M. Auv-aided localization for underwater sensor networks. In: IEEE. *Wireless Algorithms, Systems and Applications, 2007. WASA 2007. International Conference on*. [S.l.], 2007. p. 44–54. Citado na página 27.
- FRANCA, R. P. et al. Trajectory generation for bathymetry based auv navigation and localization. *IFAC-PapersOnLine*, Elsevier, v. 48, n. 16, p. 95–100, 2015. Citado na página 22.
- GARCIA, R. et al. Positioning an underwater vehicle through image mosaicking. In: IEEE. *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*. [S.l.], 2001. v. 3, p. 2779–2784. Citado na página 26.
- GORDON, N. J.; SALMOND, D. J.; SMITH, A. F. Novel approach to nonlinear/non-gaussian bayesian state estimation. In: IET. *IEE Proceedings F (Radar and Signal Processing)*. [S.l.], 1993. v. 140, n. 2, p. 107–113. Citado na página 22.

- GRENON, G. et al. Enhancement of the inertial navigation system for the morpheus autonomous underwater vehicles. *Oceanic Engineering, IEEE Journal of, IEEE*, v. 26, n. 4, p. 548–560, 2001. Citado na página 29.
- HASTINGS, W. K. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, Biometrika Trust, v. 57, n. 1, p. 97–109, 1970. Citado na página 72.
- HYAKUDOME, T. Design of autonomous underwater vehicle. *International Journal of Advanced Robotic Systems*, v. 8, n. 1, p. 122–130, 2011. Citado na página 21.
- JULIER, S. J.; UHLMANN, J. K. A new extension of the kalman filter to nonlinear systems. In: ORLANDO, FL. *Int. symp. aerospace/defense sensing, simul. and controls*. [S.l.], 1997. v. 3, n. 26, p. 3–2. Citado na página 22.
- KALMAN, R. E. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, American Society of Mechanical Engineers, v. 82, n. 1, p. 35–45, 1960. Citado na página 22.
- KARLSSON, R.; GUSFAFSSON, F.; KARLSSON, T. Particle filtering and cramer-rao lower bound for underwater navigation. In: IEEE. *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*. [S.l.], 2003. v. 6, p. VI–65. Citado na página 22.
- KIM, D. et al. Artificial landmark-based underwater localization for auvs using weighted template matching. *Intelligent Service Robotics*, Springer, p. 1–10, 2014. Citado na página 27.
- KINSEY, J. C.; EUSTICE, R. M.; WHITCOMB, L. L. A survey of underwater vehicle navigation: Recent advances and new challenges. In: *IFAC Conference of Manoeuvring and Control of Marine Craft*. [S.l.: s.n.], 2006. Citado na página 28.
- LI, C.-y.; JI, H.-b. A new particle filter with ga-mcmc resampling. In: IEEE. *Wavelet Analysis and Pattern Recognition, 2007. ICWAPR'07. International Conference on*. [S.l.], 2007. v. 1, p. 146–150. Citado na página 39.
- MARTINEZ, A. et al. Model-aided navigation with sea current estimation for an autonomous underwater vehicle. *International Journal of Advanced Robotic Systems*, InTech, v. 12, 2015. Citado na página 108.
- NAKATANI, T. et al. Terrain based localization for pinpoint observation of deep seafloors. In: IEEE. *OCEANS 2009-EUROPE*. [S.l.], 2009. p. 1–6. Citado 2 vezes nas páginas 29 e 30.
- NEMRA, A.; AOUF, N. Robust ins/gps sensor fusion for uav localization using sdre nonlinear filtering. *Sensors Journal, IEEE, IEEE*, v. 10, n. 4, p. 789–798, 2010. Citado na página 22.
- PAULL, L. et al. Auv navigation and localization: A review. *Oceanic Engineering, IEEE Journal of, IEEE*, v. 39, n. 1, p. 131–149, 2014. Citado 2 vezes nas páginas 21 e 28.
- PRÉ-SAL. 2016. <<http://http://www.petrobras.com.br/pt/nossas-atividades/areas-de-atuacao/exploracao-e-producao-de-petroleo-e-gas/pre-sal/.html>>. Acessado: 2015-01-22. Citado na página 21.

- RAUMANN, C. G. et al. *Digital Bathymetric Model of Mono Lake, California*. 2012. <<http://pubs.usgs.gov/mf/2002/2393/>>. Acessado: 2014-11-13. Citado 2 vezes nas páginas 30 e 41.
- RIGBY, P.; PIZARRO, O.; WILLIAMS, S. B. Towards geo-referenced auv navigation through fusion of usbl and dvl measurements. In: IEEE. *OCEANS 2006*. [S.l.], 2006. p. 1–6. Citado na página 29.
- SIEGWART, R.; NOURBAKHSH, I. R.; SCARAMUZZA, D. *Introduction to autonomous mobile robots*. [S.l.]: MIT press, 2011. Citado na página 23.
- SILVA, J. E. da et al. Modeling and simulation of the lauv autonomous underwater vehicle. In: *13th IEEE IFAC International Conference on Methods and Models in Automation and Robotics*. [S.l.: s.n.], 2007. Citado 2 vezes nas páginas 39 e 40.
- SMITH, S.; KRONEN, D. Experimental results of an inexpensive short baseline acoustic positioning system for auv navigation. In: IEEE. *OCEANS'97. MTS/IEEE Conference Proceedings*. [S.l.], 1997. v. 1, p. 714–720. Citado na página 28.
- STUTTERS, L. et al. Navigation technologies for autonomous underwater vehicles. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, IEEE, v. 38, n. 4, p. 581–589, 2008. Citado na página 22.
- THOMAS, H. C. Gib buoys: an interface between space and depths of the oceans. In: IEEE. *Autonomous Underwater Vehicles, 1998. AUV'98. Proceedings Of The 1998 Workshop on*. [S.l.], 1998. p. 181–184. Citado na página 21.
- THRUN, S.; BURGARD, W.; FOX, D. *Probabilistic robotics*. [S.l.]: MIT press, 2005. Citado 12 vezes nas páginas 31, 32, 34, 35, 36, 37, 38, 39, 68, 71, 72 e 77.
- TILTON, A. K.; GHIOTTO, S.; MEHTA, P. G. A comparative study of nonlinear filtering techniques. In: *FUSION*. [S.l.: s.n.], 2013. p. 1827–1834. Citado na página 37.
- VICKERY, K. Acoustic positioning systems. a practical overview of current systems. In: IEEE. *Autonomous Underwater Vehicles, 1998. AUV'98. Proceedings of the 1998 Workshop on*. [S.l.], 1998. p. 5–17. Citado 2 vezes nas páginas 21 e 28.
- WON, S.-h. et al. A kalman/particle filter-based position and orientation estimation method using a position sensor/inertial measurement unit hybrid system. *Industrial Electronics, IEEE Transactions on*, IEEE, v. 57, n. 5, p. 1787–1798, 2010. Citado na página 22.
- XU, X.; NEGAHDARIPOUR, S. Vision-based motion sensing for underwater navigation and mosaicing of ocean floor images. In: IEEE. *OCEANS'97. MTS/IEEE Conference Proceedings*. [S.l.], 1997. v. 2, p. 1412–1417. Citado na página 26.
- YOUNGBERG, J. W. *Method for extending GPS to underwater applications*. [S.l.]: Google Patents, 1992. US Patent 5,119,341. Citado na página 21.
- YUN, X. et al. Testing and evaluation of an integrated gps/ins system for small auv navigation. *Oceanic Engineering, IEEE Journal of*, IEEE, v. 24, n. 3, p. 396–404, 1999. Citado na página 29.



ZADEH, R. A.; GHOSH, A.; LEDWICH, G. Combination of kalman filter and least-error square techniques in power system. *Power Delivery, IEEE Transactions on*, IEEE, v. 25, n. 4, p. 2868–2880, 2010. Citado na página 22.

ZHANG, P. et al. Impaired-cooling-condition detection using dc-signal injection for soft-starter-connected induction motors. *Industrial Electronics, IEEE Transactions on*, IEEE, v. 56, n. 11, p. 4642–4650, 2009. Citado na página 22.

ZHANG, P.; MILIOS, E. E.; GU, J. Underwater robot localization using artificial visual landmarks. In: IEEE. *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on*. [S.l.], 2004. p. 705–710. Citado na página 21.