



AVALIAÇÃO DE ALGORITMOS BASEADOS EM FILTRO DE KALMAN  
APLICADOS EM *BEARING-ONLY TRACKING*

Danilo Monteiro Olivieri

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientadora: Mariane Rembold Petraglia

Rio de Janeiro  
Junho de 2023

AVALIAÇÃO DE ALGORITMOS BASEADOS EM FILTRO DE KALMAN  
APLICADOS EM *BEARING-ONLY TRACKING*

Danilo Monteiro Olivieri

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Orientadora: Mariane Rembold Petraglia

Aprovada por: Prof.<sup>a</sup> Mariane Rembold Petraglia, Ph.D.

Prof. Markus Vinícius Santos Lima, D.Sc.

Prof. José Antonio Apolinário Junior, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

JUNHO DE 2023

Monteiro Olivieri, Danilo

Avaliação de algoritmos baseados em Filtro de Kalman aplicados em *Bearing-only Tracking*/Danilo Monteiro Olivieri. – Rio de Janeiro: UFRJ/COPPE, 2023.

XIX, 113 p.: il.; 29, 7cm.

Orientadora: Mariane Rembold Petraglia

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2023.

Referências Bibliográficas: p. 69 – 73.

1. Filtro de Kalman.
2. *Bearing-only tracking*.
3. Guerra eletrônica. I. Rembold Petraglia, Mariane. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

*À minha esposa,  
minha maior incentivadora.*

# Agradecimentos

Meus agradecimentos são primeiramente dedicados à minha esposa, minha maior incentivadora e que não mediu esforço para me apoiar nesta jornada.

Expresso minha imensa gratidão à minha orientadora, Professora Mariane Petraglia, pelo auxílio e apoio na realização desta pesquisa e por tudo que me ensinou durante o curso de Mestrado em Engenharia Elétrica.

Agradeço também ao meu coorientador técnico da Marinha do Brasil, Jorge Costa, e demais pesquisadores e oficiais do Instituto de Pesquisas da Marinha e do Centro de Guerra Eletrônica e Acústica da Marinha, que de alguma forma contribuíram para o desenvolvimento desta pesquisa.

Registro minha gratidão também aos demais professores do Programa de Engenharia Elétrica que conduziram este Engenheiro de Produção nesta empreitada de desbravar o universo de Processamento de Sinais.

Ficam meus votos de agradecimento ao meu amigo, “campanha de Marinha” e doutorando Capitão-Tenente (EN) Diego Marques pelo companheirismo e pelas prazerosas discussões que foram fonte de aprendizado ao longo de nossas jornadas acadêmicas.

Por fim, agradeço a meus pais e irmãos pela base sólida que me permitiu chegar até aqui.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

AValiação DE ALGORITMOS BASEADOS EM FILTRO DE KALMAN  
APLICADOS EM *BEARING-ONLY TRACKING*

Danilo Monteiro Olivieri

Junho/2023

Orientadora: Mariane Rembold Petraglia

Programa: Engenharia Elétrica

Apresenta-se, nesta dissertação de Mestrado, uma avaliação de diferentes algoritmos baseados em Filtros de Kalman para o problema de rastreamento apenas por marcação (*Bearing-Only Tracking*), visando subsidiar uma implementação no sistema de apoio ao combate desenvolvido pelo Instituto de Pesquisas da Marinha. Foram estudados 9 algoritmos, sendo 7 desenvolvidos especificamente para este problema e um deles uma forma de inicialização com filtros em paralelo aplicável aos 8 demais. Os modelos foram simulados em 15 cenários, que combinaram 5 formas de inicialização do problema e 3 formas de movimentação dos navios observador e alvo, e os resultados foram comparados por métricas de erro quadrático médio de estimação de distância e velocidade do alvo (balizados pelo Limite Inferior de Cramér-Rao como referência de erro mínimo), de credibilidade das previsões com base na precisão da estimativa do erro de previsão, além de considerar as taxas de convergência de rota e estabilidade dos algoritmos. Os resultados obtidos não indicaram uma única solução dominante. Os dois modelos baseados em coordenada polar apresentaram problemas de estabilidade, tendo outros dois modelos apresentando problemas de convergência em alguns cenários e um outro com 100% de convergência, mas em patamar de erro acima dos demais. Os algoritmos que se destacaram foram *Bias-Compensated Pseudo-Linear Kalman Filter* (BC-PLKF), *Ensemble Kalman Filter* (EnKF) e *Selective Angle Measurements Instrumental Variable-Based Kalman Filter* (SAM-IVKF), que ficaram entre os melhores resultados em erro de estimativa e credibilidade em todos os cenários, sendo, então, recomendados para uma futura implementação real. Também foi indicada a inicialização com parametrização de distância e velocidade, que garantiu bons resultados de erro e credibilidade, além de robustez dos algoritmos, elevando a taxa de convergência dos que tiveram problema.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

## ASSESSMENT OF KALMAN FILTER-BASED ALGORITHMS APPLIED TO BEARING-ONLY TRACKING

Danilo Monteiro Olivieri

June/2023

Advisor: Mariane Rembold Petraglia

Department: Electrical Engineering

This master's thesis presents an evaluation of different Bearing-Only Tracking algorithms based on Kalman Filters, aiming to subsidize an implementation in the combat support system developed by Brazilian Navy's Research Institute. Nine algorithms were studied, seven of which developed specifically for this problem and one of them being a form of initialization running filters in parallel that can be applied to all of them. The models were simulated in 15 scenarios, combining 5 ways of initialization problem and 3 ways of observer and target ships movement. The results were compared regarding mean squared-error metrics for range and target velocity estimation (considering Cramér-Rao Lower Bound as minimum error reference), credibility of predictions, based on the accuracy of prediction error estimation, besides route convergence and algorithm stability rates. Results showed there was not a single dominant solution. The two models based on polar coordinates presented stability problems, two other models had convergence issue in some scenarios, and a fifth model reached 100% of convergence rate, but having higher error level. The algorithms Bias- Compensated Pseudo-Linear Kalman Filter (BC-PLKF), Ensemble Kalman Filter (EnKF) and Selective Angle Measurements Instrumental Variable-Based Kalman Filter (SAM-IVKF) stood out, reaching best overall results regarding estimation error and credibility in all scenarios and then they were recommended for a future application. In addition, initialization based on the range and velocity parameterization was indicated, which ensured superior performance in terms of error and credibility, and greater robustness of the algorithms, increasing the convergence rate of those that had a problem.

# Sumário

<b>Lista de Figuras</b>	<b>xi</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>Lista de Símbolos</b>	<b>xv</b>
<b>Lista de Abreviaturas</b>	<b>xvii</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Revisão Bibliográfica</b>	<b>4</b>
2.1 Filtros de Kalman . . . . .	4
2.1.1 Filtro de Kalman Linear . . . . .	5
2.1.2 Filtro de Kalman Estendido . . . . .	7
2.1.3 Filtro de Kalman de Conjunto . . . . .	8
2.2 Modelos de <i>Bearing-Only Tracking</i> . . . . .	10
2.2.1 Modelos Clássicos de BOT . . . . .	10
2.2.2 Modelos de BOT baseados em Filtro de Kalman . . . . .	13
<b>3 Aplicação de Filtro de Kalman ao Problema de <i>Bearing-Only Tracking</i></b>	<b>16</b>
3.1 Formulação do Espaço de Estados . . . . .	16
3.2 Aplicação Direta do Filtro de Kalman Estendido . . . . .	19
3.3 Filtro de Kalman Pseudolinear . . . . .	20
3.4 Filtro de Kalman Pseudolinear com Compensação de Viés . . . . .	22
3.5 Filtro de Kalman baseado em Variáveis Instrumentais e Medidas de Ângulo Seletivas . . . . .	25
3.6 Aplicação Direta do Filtro de Kalman de Conjunto . . . . .	27
3.7 Filtro de Kalman de Conjunto Aumentado . . . . .	27
3.8 Filtro de Kalman Estendido com Coordenada Polar Modificada . . . . .	32
3.9 Filtro de Kalman Estendido com Coordenada Log-Polar . . . . .	37

3.10	Filtro de Kalman Estendido Aplicado a BOT com Parametrização de Distância e Velocidade . . . . .	41
<b>4</b>	<b>Preparação dos Experimentos</b>	<b>45</b>
4.1	Metodologia . . . . .	45
4.2	Cenários Simulados . . . . .	46
4.2.1	Inicialização . . . . .	46
4.2.2	Movimentação dos Navios . . . . .	47
4.2.3	Demais Parâmetros . . . . .	50
4.3	Métricas de Desempenho . . . . .	50
4.3.1	Raiz-Quadrada do Erro Quadrático Médio . . . . .	51
4.3.2	Limite Inferior de Cramér-Rao Posterior . . . . .	52
4.3.3	Erro de Estimção Quadrático Normalizado Médio . . . . .	53
4.3.4	Tempo de Processamento . . . . .	53
4.3.5	Estabilidade e Convergência . . . . .	54
<b>5</b>	<b>Resultados e Discussão</b>	<b>56</b>
5.1	Desempenho Geral dos Modelos . . . . .	56
5.2	Impacto da Inicialização . . . . .	61
5.3	Impacto da Movimentação dos Navios . . . . .	64
5.4	Tempo de Processamento . . . . .	65
<b>6</b>	<b>Conclusão</b>	<b>67</b>
	<b>Referências Bibliográficas</b>	<b>69</b>
<b>A</b>	<b>Linearização da Equação de Estados do LP-EKF</b>	<b>74</b>
<b>B</b>	<b>Códigos em MATLAB</b>	<b>77</b>
B.1	EKF_loop.m . . . . .	77
B.2	PLKF_loop.m . . . . .	78
B.3	BCPLKF_loop.m . . . . .	79
B.4	SAMIVKF_loop.m . . . . .	80
B.5	EnKF_loop.m . . . . .	81
B.6	AEnKF_loop.m . . . . .	82
B.7	MPEKF_loop.m . . . . .	83
B.8	LPEKF_loop.m . . . . .	84
B.9	RP_EKF_loop.m . . . . .	86
B.10	RP_PLKF_loop.m . . . . .	87
B.11	RP_BCPLKF_loop.m . . . . .	88
B.12	RP_SAMIVKF_loop.m . . . . .	90

B.13	RP_EnKF_loop.m . . . . .	91
B.14	RP_AEnKF_loop.m . . . . .	92
B.15	RP_MPEKF_loop.m . . . . .	93
B.16	RP_LPEKF_loop.m . . . . .	94
B.17	AtualizaPesosRP.m . . . . .	95
B.18	arctan_BOT.m . . . . .	96
<b>C</b>	<b>Resultados Detalhados por Cenário</b>	<b>98</b>
C.1	Movimentação 1, Inicialização 1 . . . . .	98
C.2	Movimentação 1, Inicialização 2 . . . . .	99
C.3	Movimentação 1, Inicialização 3 . . . . .	100
C.4	Movimentação 1, Inicialização 4 . . . . .	101
C.5	Movimentação 1, Inicialização 5 . . . . .	102
C.6	Movimentação 2, Inicialização 1 . . . . .	103
C.7	Movimentação 2, Inicialização 2 . . . . .	104
C.8	Movimentação 2, Inicialização 3 . . . . .	105
C.9	Movimentação 2, Inicialização 4 . . . . .	106
C.10	Movimentação 2, Inicialização 5 . . . . .	107
C.11	Movimentação 3, Inicialização 1 . . . . .	108
C.12	Movimentação 3, Inicialização 2 . . . . .	109
C.13	Movimentação 3, Inicialização 3 . . . . .	110
C.14	Movimentação 3, Inicialização 4 . . . . .	111
C.15	Movimentação 3, Inicialização 5 . . . . .	112

# Lista de Figuras

2.1	Exemplo de tática com duas pernadas . . . . .	12
2.2	Principais algoritmos baseados em Filtro de Kalman . . . . .	15
3.1	Representação da análise de BOT . . . . .	17
3.2	Diagrama de inicialização dos filtros . . . . .	41
4.1	Movimentação 1 . . . . .	48
4.2	Movimentação 2 . . . . .	49
4.3	Movimentação 3 . . . . .	50
5.1	BC-PLKF – $M_2I_4$ – Rastreamento . . . . .	59
5.2	BC-PLKF – $M_2I_4$ – Distância e Velocidade do alvo . . . . .	59
5.3	SAM-IVKF – $M_2I_3$ – Rastreamento . . . . .	60
5.4	SAM-IVKF – $M_2I_3$ – Distância e Velocidade do alvo . . . . .	60
5.5	EnKF – $M_3I_3$ – Rastreamento . . . . .	60
5.6	EnKF – $M_3I_3$ – Distância e Velocidade do alvo . . . . .	61
5.7	BC-PLKF – $M_1$ – Métricas . . . . .	62
5.8	BC-PLKF – $M_2$ – Métricas . . . . .	62
5.9	BC-PLKF – $M_3$ – Métricas . . . . .	63
5.10	BC-PLKF – $M_1I_1$ – Distância e Velocidade do alvo . . . . .	63
5.11	BC-PLKF – $M_1I_5$ – Distância e Velocidade do alvo . . . . .	64
5.12	SAM-IVKF – $M_2$ – Métricas . . . . .	65
5.13	EKF – $M_3$ – Métricas . . . . .	66
C.1	Métricas – $M_1I_1$ . . . . .	99
C.2	Métricas – $M_1I_2$ . . . . .	100
C.3	Métricas – $M_1I_3$ . . . . .	101
C.4	Métricas – $M_1I_4$ . . . . .	102
C.5	Métricas – $M_1I_5$ . . . . .	103
C.6	Métricas – $M_2I_1$ . . . . .	104
C.7	Métricas – $M_2I_2$ . . . . .	105
C.8	Métricas – $M_2I_3$ . . . . .	106

C.9 Métricas – $M_2I_4$ . . . . .	107
C.10 Métricas – $M_2I_5$ . . . . .	108
C.11 Métricas – $M_3I_1$ . . . . .	109
C.12 Métricas – $M_3I_2$ . . . . .	110
C.13 Métricas – $M_3I_3$ . . . . .	111
C.14 Métricas – $M_3I_4$ . . . . .	112
C.15 Métricas – $M_3I_5$ . . . . .	113

# Lista de Tabelas

2.1	Algoritmo do Filtro de Kalman Linear . . . . .	7
2.2	Algoritmo do Filtro de Kalman Estendido . . . . .	8
2.3	Algoritmo do Filtro de Kalman de Conjunto . . . . .	11
3.1	Filtro de Kalman Estendido aplicado a BOT . . . . .	19
3.2	Filtro de Kalman Pseudolinear . . . . .	21
3.3	Filtro de Kalman Pseudolinear com Compensação de Viés . . . . .	25
3.4	Filtro de Kalman baseado em Variáveis Instrumentais e Medidas de Ângulo Seletivas . . . . .	27
3.5	Filtro de Kalman de Conjunto aplicado a BOT . . . . .	28
3.6	Coefficientes da expansão de caos polinomial arbitrária . . . . .	30
3.7	Algoritmo geral do Filtro de Kalman de Conjunto Aumentado . . . . .	33
3.8	Filtro de Kalman Estendido com Coordenada Polar Modificada . . . . .	38
3.9	Filtro de Kalman Estendido com Coordenada Log-Polar . . . . .	40
3.10	Filtro de Kalman Estendido com Parametrização de Distância e Ve- locidade . . . . .	44
4.1	Parâmetros para inicialização por cenário . . . . .	47
5.1	Razão média entre PCRLB e valores reais por cenário de movimentação	56
5.2	Número de cenários por <i>ranking</i> das métricas . . . . .	58
C.1	Métricas – $M_1I_1$ (médias temporais) . . . . .	98
C.2	Métricas – $M_1I_2$ (médias temporais) . . . . .	99
C.3	Métricas – $M_1I_3$ (médias temporais) . . . . .	100
C.4	Métricas – $M_1I_4$ (médias temporais) . . . . .	101
C.5	Métricas – $M_1I_5$ (médias temporais) . . . . .	102
C.6	Métricas – $M_2I_1$ (médias temporais) . . . . .	103
C.7	Métricas – $M_2I_2$ (médias temporais) . . . . .	104
C.8	Métricas – $M_2I_3$ (médias temporais) . . . . .	105
C.9	Métricas – $M_2I_4$ (médias temporais) . . . . .	106
C.10	Métricas – $M_2I_5$ (médias temporais) . . . . .	107

C.11 Métricas – $M_3I_1$ (médias temporais) . . . . .	108
C.12 Métricas – $M_3I_2$ (médias temporais) . . . . .	109
C.13 Métricas – $M_3I_3$ (médias temporais) . . . . .	110
C.14 Métricas – $M_3I_4$ (médias temporais) . . . . .	111
C.15 Métricas – $M_3I_5$ (médias temporais) . . . . .	112

# Lista de Símbolos

$(\tilde{\cdot})$	Variável medida (contendo incerteza), p. 18
$\alpha_t, \alpha_o$	Rumo do alvo e do observador, respectivamente, p. 11, 16
$\beta$	Marcação ( <i>bearing</i> ), p. 11
$\dot{\beta}, \ddot{\beta}$	Primeira e segunda derivadas de $\beta$ em relação ao tempo, respectivamente, p. 32
$\eta$	Ruído de medição da marcação, p. 18
$(\hat{\cdot})$	Variável calculada / estimada, p. 5
$\mathbb{E}[\cdot]$	Operador de valor esperado, p. 5
$\mathbf{0}$	Vetor nulo ou matriz nula, p. 5
$\mathbf{I}$	Matriz Identidade, p. 6
$\mathbf{a}_o, a_{o_x}, a_{o_y}$	Aceleração do observador (vetor, componente $x$ , componente $y$ ), p. 17
$\mathbf{a}_t, a_{t_x}, a_{t_y}$	Aceleração do alvo (vetor, componente $x$ , componente $y$ ), p. 16
$\mathbf{s}_o, s_{o_x}, s_{o_y}$	Posição do observador no plano cartesiano $xy$ (vetor, componente $x$ , componente $y$ ), p. 17
$\mathbf{s}_t, s_{t_x}, s_{t_y}$	Posição do alvo no plano cartesiano $xy$ (vetor, componente $x$ , componente $y$ ), p. 16
$\mathbf{u}$	Perturbação na trajetória do alvo, p. 22
$\mathcal{N}(\cdot, \cdot)$	Distribuição Normal (média, variância), p. 8
$\sigma_\eta$	Desvio-padrão do ruído de medição da marcação, p. 18
$\tau$	Período de amostragem, p. 17

$\xi_i$	Fator de ponderação do $i$ -ésimo filtro do RP-EKF, p. 42
$a_{o_{\text{tan}}}, a_{o_{\text{rad}}}$	Aceleração tangencial e radial do navio-observador, respectivamente, p. 32
$k_m$	Instante amostral $k$ imediatamente após a primeira manobra do navio-observador, p. 51
$r, \mathbf{r}, r_x, r_y$	Distância do observador ao alvo ( <i>range</i> ) (módulo, vetor, componente $x$ , componente $y$ ), p. 11
$v_o, \mathbf{v}_o, v_{o_x}, v_{o_y}$	Velocidade do observador (módulo, vetor, componente $x$ , componente $y$ ), p. 11
$v_t, \mathbf{v}_t, v_{t_x}, v_{t_y}$	Velocidade do alvo (módulo, vetor, componente $x$ , componente $y$ ), p. 11
$[\cdot]^T$	Operador de transposição de matriz ou vetor, p. 5
$[\cdot]^{-1}$	Operador de Matriz Inversa, p. 6
$\text{diag}[\cdot]$	Matriz diagonal, p. 37
nmi	Símbolo da unidade milha náutica (ou marítima) comumente adotado ( <i>nautical mile</i> ), p. 36

# Lista de Abreviaturas

AEnKF	Filtro de Kalman de Conjunto Aumentado ( <i>Augmented Ensemble Kalman Filter</i> ), p. 14
ANEEES	Erro de Estimação Quadrático Normalizado Médio ( <i>Average Normalized Estimation Error Squared</i> ), p. 53
BC-PLKF	Filtro de Kalman Pseudolinear com Compensação de Viés ( <i>Bias-Compensated Pseudo-Linear Kalman Filter</i> ), p. 14
BOTMA	Análise de Movimento do Alvo Apenas por Marcação ( <i>Bearing-only Target Motion Analysis</i> ), p. 2
BOT	Rastreamento Apenas por Marcação ( <i>Bearing-only Tracking</i> ), p. 2
CONSET	Condição de Silêncio Eletrônico, p. 1
CRLB	Limite Inferior de Cramér-Rao ( <i>Cramér-Rao Lower Bound</i> ), p. 14
EKF	Filtro de Kalman Estendido ( <i>Extended Kalman Filter</i> ), p. 7
EnKF	Filtro de Kalman de Conjunto ( <i>Ensemble Kalman Filter</i> ), p. 8
GBC-PLKF	Filtro de Kalman Pseudolinear com Compensação de Viés Generalizado ( <i>Generalized Bias-Compensated Pseudo-Linear Kalman Filter</i> ), p. 14
GWN	Ruído Branco Gaussiano ( <i>Gaussian White Noise</i> ), p. 18
IPqM	Instituto de Pesquisas da Marinha, p. 2
IV	Variáveis Instrumentais ( <i>Instrumental Variables</i> ), p. 25
LKF	Filtro de Kalman Linear ( <i>Linear Kalman Filter</i> ), p. 6

LMMSE	Mínimo Erro Quadrático Médio Linear ( <i>Linear Minimum Mean Squared Error</i> ), p. 28
LMMSE	Mínimo Erro Quadrático Médio Linear Generalizado ( <i>Generalized Linear Minimum Mean Squared Error</i> ), p. 28
LP-EKF	Filtro de Kalman Estendido com Coordenada Log-Polar ( <i>Log-Polar Coordinate Extended Kalman Filter</i> ), p. 14
LPT	Tempo de Processamento do <i>Loop</i> ( <i>Loop Processing Time</i> ), p. 53
MAGE	Medidas de Apoio à Guerra Eletrônica, p. 2
MP-EKF	Filtro de Kalman Estendido com Coordenada Polar Modificada ( <i>Modified Polar Coordinate Extended Kalman Filter</i> ), p. 13
MRU	Movimento Retilíneo Uniforme, p. 10
MSE	Erro quadrático médio ( <i>Mean-Squared Error</i> ), p. 4
PCRLB	Limite Inferior de Cramér-Rao Posterior ( <i>Posterior Cramér-Rao Lower Bound</i> ), p. 52
PDF	Função Densidade de Probabilidade ( <i>Probability Density Function</i> ), p. 8
PLKF	Filtro de Kalman Pseudolinear ( <i>Pseudo-Linear Kalman Filter</i> ), p. 13
RLS	<i>Recursive Least-Squares</i> , p. 6
RMSE	Raiz-Quadrada do Erro Quadrático Médio ( <i>Root-Mean-Squared Error</i> ), p. 51
RP-LP-EKF	Filtro de Kalman Estendido com Coordenada Log-Polar e Parametrização de Distância ( <i>Range Parameterized Log-Polar Coordinate Extended Kalman Filter</i> ), p. 14
RP-MP-EKF	Filtro de Kalman Estendido com Coordenada Polar Modificada e Parametrização de Distância ( <i>Range Parameterized Modified Polar Coordinate Extended Kalman Filter</i> ), p. 14
RPT	Tempo de Processamento Relativo ( <i>Relative Processing Time</i> ), p. 53
RSE	Raiz do Erro Quadrático ( <i>Root-Squared Error</i> ), p. 54

SAM-IVKF	Filtro de Kalman baseado em Variáveis Instrumentais e Medidas de Ângulo Seletivas ( <i>Selective Angle Measurements Instrumental Variable-Based Kalman Filter</i> ), p. 14
SR	Taxa de Estabilidade ( <i>Stability Rate</i> ), p. 54
TCR	Taxa de Convergência de Rota ( <i>Track Convergence Rate</i> ), p. 55
TMA	Análise de Movimento do Alvo ( <i>Target Motion Analysis</i> ), p. 2
TPT	Tempo de Processamento Total ( <i>Total Processing Time</i> ), p. 53
UC	Conversão Descorrelacionada ( <i>Uncorrelated Conversion</i> ), p. 29

# Capítulo 1

## Introdução

O Brasil depende economicamente de sua Amazônia Azul. Os modais marítimo, fluvial e lacustre concentraram 83,4% em valor (97,3% em toneladas) do comércio exterior em 2022, de acordo com os dados do Ministério da Indústria, Comércio Exterior e Serviços [1]. Somado a isso, 94,7% em volume da produção nacional de petróleo e gás natural de 2022 foi *offshore*, segundo dados da Agência Nacional de Petróleo, Gás Natural e Biocombustíveis [2]. Para que estes resultados ocorram, é necessário que haja segurança para a realização destas atividades, o que é garantido pela capacidade de dissuasão do poder naval brasileiro.

Um dos fatores de sucesso das ações de defesa é operar com discrição, ou seja, sem revelar sua posição até que se efetue uma medida ofensiva. Considerando os meios de superfície, estes operam em Condição de Silêncio Eletrônico (CONSET), usando sensores passivos para detectar presença de sinais de radares ou sonares emitidos por outros navios. Além da vantagem de minimizar a exposição para não ser facilmente encontrado quando operando em CONSET, de modo geral, o emprego de técnicas de detecção passiva permite identificar emissores a uma distância maior, uma vez que o decaimento da potência do sinal emitido é proporcional ao quadrado da distância, ao passo que um radar detecta o sinal refletido, que apresenta decaimento proporcional à quarta potência da distância.

Por outro lado, a detecção passiva fornece apenas a potência do sinal observado e a sua marcação pelo cálculo do ângulo de chegada, não havendo informação de localização da fonte ou mesmo de velocidade ou rumo do emissor. Em navegação, define-se marcação como a direção de um objeto em terra ou no mar, obtida pelo navio, sendo expressa pelo ângulo no plano horizontal medido a partir do Norte (magnético ou verdadeiro) até a linha de visada que fazemos do navio, que está no vértice deste ângulo, a um objeto qualquer, medido de 000 a 360 graus no sentido dos ponteiros do relógio, e rumo como a direção de navegação de um navio, expressa pelo ângulo no plano horizontal medido a partir do Norte (magnético ou verdadeiro) até a direção para a qual aponta a proa do navio, medido da mesma forma que na

marcação [3]. Além disso, os termos “localização” e “posição” do alvo serão usados para se referir à informação de coordenadas do alvo ou mesmo distância do alvo ao observador, que, combinada à marcação, forneceria as coordenadas.

Sendo assim, para se obter as informações de posição e velocidade do emissor do sinal, o navio-observador, que é um sensor único, precisa realizar algumas manobras a fim de emular um *array* de sensores, obtendo a marcação do alvo em diferentes instantes e, por meio de algoritmo para análise de movimento relativo, estimar tais parâmetros, o que é conhecido por *Bearing-only Tracking* (BOT – rastreamento apenas por marcação), *Bearing-only Target Motion Analysis* (BOTMA – análise de movimento do alvo apenas por marcação), ou apenas *Target Motion Analysis* (TMA – análise de movimento do alvo).

MAGE (Medidas de Apoio à Guerra Eletrônica) é o sistema operativo dos navios de superfície que compreende um sensor passivo, que abrange o espectro eletromagnético na faixa de 2 GHz a 18 GHz, ligado a um sistema de suporte, o qual utiliza informações de um banco de dados para classificação dos sinais identificados (tipo de radar e navio de origem – “amigo” / “inimigo” / “neutro”). Como a localização da fonte não pode ser definida apenas com sensor passivo, é necessário estabelecer um algoritmo (ou conjunto de algoritmos) de BOT que possa suportar a estimação da localização de outros navios e potenciais ameaças.

Neste contexto, esta pesquisa de mestrado, enquadrada no Plano de Capacitação de Pessoal da Marinha, visa subsidiar o Instituto de Pesquisas da Marinha (IPqM) na evolução do MAGE Defensor, desenvolvido por seus engenheiros, com a implementação de algoritmos de BOT para navios de superfície. A abordagem inicial escolhida pelo IPqM consiste na aplicação de Filtros de Kalman para a solução, uma vez que se trata de uma técnica com ampla referência bibliográfica e, no geral, com baixa demanda de processamento computacional. Com isso, o objetivo desta dissertação é avaliar soluções baseadas em Filtro de Kalman que possam ser satisfatórias para serem aplicadas ao MAGE e compreenderá:

- i. revisão bibliográfica das técnicas e práticas atuais de aplicação de Filtro de Kalman ao problema de BOT, estabelecendo uma seleção das mais relevantes e/ou adequadas para implementação;
- ii. modelagem da manobra do navio, com incertezas nas medições das marcações;
- iii. implementação e simulação computacional dos algoritmos, com comparação de desempenho na precisão das estimativas e no tempo de processamento; e
- iv. discussão dos resultados e recomendação dos modelos mais adequados às necessidades do IPqM.

Para isso, esta dissertação está estruturada da seguinte forma: o Capítulo 2 contém a revisão bibliográfica sobre Filtro de Kalman e uma visão consolidada da bibliografia sobre o problema de BOT; o Capítulo 3 descreve a aplicação de Filtro de Kalman ao problema de BOT, com detalhamento dos modelos levantados, assim como suas características e inovações aportadas; o Capítulo 4 apresenta a metodologia utilizada para comparação dos modelos, o que inclui a implementação computacional, cenários de teste e métricas de desempenho; o Capítulo 5 discute os resultados das simulações e o desempenho dos modelos, tanto de forma geral quanto sob os pontos de vista do tipo de inicialização e do tipo de manobra; e o Capítulo 6 conclui o trabalho com as considerações finais e recomendações pertinentes.

# Capítulo 2

## Revisão Bibliográfica

Para esta dissertação, foi realizada uma pesquisa sobre a fundamentação teórica de Filtro de Kalman e, também, sobre soluções para o problema de BOT. Dentre as soluções, foram identificados dois grandes grupos: os algoritmos baseados em uma análise geométrica e os modelos com base em algoritmos adaptativos ou otimizantes, com maior destaque para soluções baseadas em Filtro de Kalman, as quais foram avaliadas para se identificar as principais mudanças e melhorias propostas, assim como os resultados práticos.

### 2.1 Filtros de Kalman

Os Filtros de Kalman têm o objetivo de minimizar o erro quadrático médio (MSE – *Mean-Squared Error*) de sinais não-estacionários ruidosos, permitindo uma estimativa indireta de variáveis de estado de sistemas dinâmicos. Comparado ao Filtro de Wiener, que fornece a solução de mínimo MSE para os parâmetros desconhecidos em um ambiente de sinal estacionário no sentido amplo, o Filtro de Kalman fornece a solução MSE ótima para ambientes variantes no tempo, envolvendo sistemas dinâmicos lineares com ruídos não-estacionários e sinais com formulação estocástica [4].

Para isso, é necessário definir o espaço de estados que descreve o sistema, onde os estados desconhecidos podem ser escritos como uma combinação linear dos estados estimados anteriormente e dos sinais medidos mais atuais. Sendo  $\mathbf{x}$  o vetor com as variáveis de estado que se deseja estimar,  $\mathbf{y}$  o vetor com os sinais medidos, as equações do espaço de estados de um sistema discreto no tempo e variante no tempo, para um instante  $k$ , podem ser escritas como em (2.1).

$$\begin{cases} \mathbf{x}(k) = \mathbf{A}(k-1)\mathbf{x}(k-1) + \mathbf{B}(k)\mathbf{n}(k) \\ \mathbf{y}(k) = \mathbf{C}^T(k)\mathbf{x}(k) + \mathbf{n}_1(k) \end{cases} \quad (2.1)$$

As matrizes  $\mathbf{A}(k-1)$  e  $\mathbf{B}(k)$  representam, respectivamente, a matriz de transição de estados e a matriz do processo gerador das variáveis de estado a partir do ruído de excitação  $\mathbf{n}(k)$ . Já  $\mathbf{C}^T(k)$  representa a relação linear entre os estados e as saídas, somadas ao vetor de ruído  $\mathbf{n}_1(k)$ . Neste caso, conforme DINIZ [4],  $\mathbf{n}_1(k)$  é multiplicado por uma matriz identidade para não ser considerada uma relação *feed-forward* entre este ruído e a saída, representando, assim, apenas uma componente estocástica da medição, o que também é adotado por AIDALA ([5] e [6]) e pelos modelos derivados deste.

Em muitos casos, o espaço de estados pode não ter uma relação linear entre as variáveis, sendo a transição de estados e/ou a saída representadas por funções não-lineares das variáveis de estado. Sendo assim, os algoritmos estudados buscam uma forma de linearizar a relação, o que será abordado posteriormente.

### 2.1.1 Filtro de Kalman Linear

O algoritmo de filtragem de Kalman linear pressupõe que as matrizes  $\mathbf{A}(k-1)$ ,  $\mathbf{B}(k)$  e  $\mathbf{C}^T(k)$  sejam conhecidas, assim como as matrizes de correlação dos ruídos do sistema, denotadas por  $\mathbf{P}_n(k)$  e  $\mathbf{P}_{n_1}(k)$ , onde  $\mathbf{P}_n(k) = \mathbb{E}[\mathbf{n}(k)\mathbf{n}^T(k)]$  e  $\mathbf{P}_{n_1}(k) = \mathbb{E}[\mathbf{n}_1(k)\mathbf{n}_1^T(k)]$ . Além disso, os ruídos devem ser processos estacionários brancos de média zero ( $\mathbb{E}[\mathbf{n}(k)] = \mathbf{0}$  e  $\mathbb{E}[\mathbf{n}_1(k)] = \mathbf{0}$ ) e não-correlacionados ( $\mathbb{E}[\mathbf{n}(k)\mathbf{n}_1^T(k)] = \mathbf{0}$ ).

A primeira etapa consiste na predição dos estados com base no estado do instante anterior. Sendo  $\hat{\mathbf{x}}$  o valor estimado de  $\mathbf{x}$  aplicado à primeira equação de (2.1), o estimador sem viés de  $\mathbf{x}$ , seu erro de medição e a matriz de correlação do erro são dados, respectivamente, por [4]:

$$\hat{\mathbf{x}}(k|k-1) = \mathbf{A}(k-1)\hat{\mathbf{x}}(k-1|k-1) \quad (2.2)$$

$$\begin{aligned} \mathbf{e}(k|k-1) &= \mathbf{x}(k) - \hat{\mathbf{x}}(k|k-1) \\ &= \mathbf{A}(k-1)\mathbf{e}(k-1|k-1) + \mathbf{B}(k)\mathbf{n}(k) \end{aligned} \quad (2.3)$$

$$\begin{aligned} \mathbf{P}_e(k|k-1) &= \mathbb{E}[\mathbf{e}(k|k-1)\mathbf{e}^T(k|k-1)] \\ &= \mathbf{A}(k-1)\mathbf{P}_e(k-1|k-1)\mathbf{A}^T(k-1) + \mathbf{B}(k)\mathbf{P}_n(k)\mathbf{B}^T(k) \end{aligned} \quad (2.4)$$

A segunda parte consiste na predição dos estados com inovação. O novo valor estimado é composto de uma parcela da estimativa  $\hat{\mathbf{x}}(k|k-1)$  e outra da medição  $\mathbf{y}(k)$  ponderadas por fatores de ganho, chamados de Ganho de Kalman. O novo estimador e seu erro são dados, respectivamente, por:

$$\hat{\mathbf{x}}(k|k) = \mathbf{G}_1(k)\hat{\mathbf{x}}(k|k-1) + \mathbf{G}(k)\mathbf{y}(k) \quad (2.5)$$

$$\begin{aligned}
\mathbf{e}(k|k) &= \mathbf{x}(k) - \hat{\mathbf{x}}(k|k) \\
&= \mathbf{x}(k) - \mathbf{G}_1(k)\hat{\mathbf{x}}(k|k-1) - \mathbf{G}(k)\mathbf{y}(k)
\end{aligned} \tag{2.6}$$

Sendo  $\hat{\mathbf{x}}(k|k)$  um estimador sem viés ( $\mathbb{E}[\mathbf{e}(k|k)] = \mathbf{0}$ ), é possível relacionar os ganhos  $\mathbf{G}_1(k)$  e  $\mathbf{G}(k)$  de forma a trabalhar com um valor único de ganho  $\mathbf{G}(k)$ , conforme (2.7).

$$\mathbf{G}_1(k) = \mathbf{I} - \mathbf{G}(k)\mathbf{C}^T(k) \tag{2.7}$$

Aplicando a segunda equação de (2.1) e as equações (2.3), (2.4) e (2.7) nas equações (2.5) e (2.6), podemos reescrever as equações de  $\hat{\mathbf{x}}(k|k)$  e de  $\mathbf{e}(k|k)$ , como também definir a matriz de correlação de  $\mathbf{e}(k|k)$ ,  $\mathbf{P}_e(k|k)$ , da seguinte forma:

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{G}(k) [\mathbf{y}(k) - \mathbf{C}^T(k)\hat{\mathbf{x}}(k|k-1)] \tag{2.8}$$

$$\mathbf{e}(k|k) = [\mathbf{I} - \mathbf{G}(k)\mathbf{C}^T(k)] \mathbf{e}(k|k-1) - \mathbf{G}(k)\mathbf{n}_1(k) \tag{2.9}$$

$$\begin{aligned}
\mathbf{P}_e(k|k) &= \mathbb{E}[\mathbf{e}(k|k)\mathbf{e}^T(k|k)] \\
&= [\mathbf{I} - \mathbf{G}(k)\mathbf{C}^T(k)] \mathbf{P}_e(k|k-1) [\mathbf{I} - \mathbf{G}(k)\mathbf{C}^T(k)]^T + \mathbf{G}(k)\mathbf{P}_{n_1}(k)\mathbf{G}^T(k) \\
&= [\mathbf{I} - \mathbf{G}(k)\mathbf{C}^T(k)] \mathbf{P}_e(k|k-1) \\
&\quad - \{[\mathbf{I} - \mathbf{G}(k)\mathbf{C}^T(k)] \mathbf{P}_e(k|k-1)\mathbf{C}(k) - \mathbf{G}(k)\mathbf{P}_{n_1}(k)\} \mathbf{G}^T(k)
\end{aligned} \tag{2.10}$$

O valor ótimo de  $\mathbf{G}(k)$  é obtido ao minimizar o traço da matriz de correlação do erro de predição  $\mathbf{P}_e(k|k)$ , que é dado em (2.10). Como demonstrado em [4], para este valor de ganho, o termo entre chaves em (2.10) é nulo, permitindo simplificar a equação de  $\mathbf{P}_e(k|k)$  para (2.12).

$$\mathbf{G}(k) = \mathbf{P}_e(k|k-1)\mathbf{C}(k) [\mathbf{C}^T(k)\mathbf{P}_e(k|k-1)\mathbf{C}(k) + \mathbf{P}_{n_1}(k)]^{-1} \tag{2.11}$$

$$\mathbf{P}_e(k|k) = [\mathbf{I} - \mathbf{G}(k)\mathbf{C}^T(k)] \mathbf{P}_e(k|k-1) \tag{2.12}$$

Com isso, definindo as condições iniciais  $\hat{\mathbf{x}}(0|0)$  e  $\mathbf{P}_e(0|0)$ , é possível estabelecer um algoritmo iterativo com a sequência de equações (2.2), (2.4), (2.11), (2.8) e (2.12), sumarizado na Tabela 2.1.

Em [4] são propostos os valores iniciais:  $\hat{\mathbf{x}}(0|0) = \mathbb{E}[\mathbf{x}(0)]$ , se conhecido, ou  $\hat{\mathbf{x}}(0|0) = \mathbf{x}(0)$ , se disponível, e também  $\mathbf{P}_e(0|0) = \mathbb{E}[\mathbf{x}(0)\mathbf{x}^T(0)]$ , se conhecido, ou  $\mathbf{P}_e(0|0) = \mathbf{x}(0)\mathbf{x}^T(0)$ , se disponível. No entanto, na Subseção 2.2.2 será visto que alguns modelos levantados propõem alternativas que sejam mais viáveis em uma situação prática, como em [7] e [6].

Uma abordagem baseada no algoritmo *Recursive Least-Squares* (RLS) pode ser usada para a dedução do Filtro de Kalman, também apresentada em [4] e aplicada em [8].

Tabela 2.1: Algoritmo do Filtro de Kalman Linear

---

**Filtro de Kalman Linear (LKF)**

---

$\hat{\mathbf{x}}(0|0)$  e  $\mathbf{P}_e(0|0)$  estimados

Para  $k > 0$ :

$$\begin{aligned}\hat{\mathbf{x}}(k|k-1) &= \mathbf{A}(k-1)\hat{\mathbf{x}}(k-1|k-1) \\ \mathbf{P}_e(k|k-1) &= \mathbf{A}(k-1)\mathbf{P}_e(k-1|k-1)\mathbf{A}^T(k-1) + \mathbf{B}(k)\mathbf{P}_n(k)\mathbf{B}^T(k) \\ \mathbf{G}(k) &= \mathbf{P}_e(k|k-1)\mathbf{C}(k) [\mathbf{C}^T(k)\mathbf{P}_e(k|k-1)\mathbf{C}(k) + \mathbf{P}_{n_1}(k)]^{-1} \\ \hat{\mathbf{x}}(k|k) &= \hat{\mathbf{x}}(k|k-1) + \mathbf{G}(k) [\mathbf{y}(k) - \mathbf{C}^T(k)\hat{\mathbf{x}}(k|k-1)] \\ \mathbf{P}_e(k|k) &= [\mathbf{I} - \mathbf{G}(k)\mathbf{C}^T(k)] \mathbf{P}_e(k|k-1)\end{aligned}$$


---

### 2.1.2 Filtro de Kalman Estendido

No caso em que o espaço de estados seja definido por equações não-lineares, o algoritmo pode ser adaptado, sendo conhecido como Filtro de Kalman Estendido (EKF – *Extended Kalman Filter*). Dadas as equações do espaço de estados

$$\begin{cases} \mathbf{x}(k) = \mathbf{f}[\mathbf{x}(k-1)] + \mathbf{B}(k)\mathbf{n}(k) \\ \mathbf{y}(k) = \mathbf{h}[\mathbf{x}(k)] + \mathbf{n}_1(k) \end{cases} \quad (2.13)$$

e sendo  $\mathbf{f}[\mathbf{x}(k-1)]$  e  $\mathbf{h}[\mathbf{x}(k)]$  conhecidas, com as mesmas premissas anteriores é possível redefinir as equações do algoritmo, seguindo os passos abaixo [4]:

i. equação (2.2) é substituída por:

$$\hat{\mathbf{x}}(k|k-1) = \mathbf{f}[\hat{\mathbf{x}}(k-1|k-1)] ; \quad (2.14)$$

ii. define-se  $\mathbf{A}(k-1)$  como:

$$\mathbf{A}(k-1) = \left. \frac{\partial \mathbf{f}[\mathbf{x}(k-1)]}{\partial \mathbf{x}(k-1)} \right|_{\mathbf{x}(k-1)=\hat{\mathbf{x}}(k-1|k-1)} \quad (2.15)$$

e se aplica em (2.4);

iii. define-se  $\mathbf{C}(k)$  como:

$$\mathbf{C}(k) = \left. \frac{\partial \mathbf{h}[\mathbf{x}(k)]}{\partial \mathbf{x}(k)} \right|_{\mathbf{x}(k)=\hat{\mathbf{x}}(k|k-1)} \quad (2.16)$$

e se aplica em (2.11); e

iv. equação (2.8) é substituída por:

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{G}(k) \{\mathbf{y}(k) - \mathbf{h}[\hat{\mathbf{x}}(k|k-1)]\} \quad (2.17)$$

e se aplica em (2.12).

Estas aproximações das matrizes  $\mathbf{A}(k-1)$  e  $\mathbf{C}(k)$  pelas primeiras derivadas das funções das variáveis de estado (como uma aproximação de Taylor de primeira ordem) e seus impactos são discutidos em alguns dos modelos de BOT estudados, que serão introduzidos na sequência. O resumo deste algoritmo é apresentado na Tabela 2.2.

Tabela 2.2: Algoritmo do Filtro de Kalman Estendido

<b>Filtro de Kalman Estendido (EKF)</b>
$\hat{\mathbf{x}}(0 0)$ e $\mathbf{P}_e(0 0)$ estimados
Para $k > 0$ :
$\hat{\mathbf{x}}(k k-1) = \mathbf{f}[\hat{\mathbf{x}}(k-1 k-1)]$
$\mathbf{A}(k-1) = \left. \frac{\partial \mathbf{f}[\mathbf{x}(k-1)]}{\partial \mathbf{x}(k-1)} \right _{\mathbf{x}(k-1) = \hat{\mathbf{x}}(k-1 k-1)}$
$\mathbf{P}_e(k k-1) = \mathbf{A}(k-1)\mathbf{P}_e(k-1 k-1)\mathbf{A}^T(k-1) + \mathbf{B}(k)\mathbf{P}_n(k)\mathbf{B}^T(k)$
$\mathbf{C}(k) = \left. \frac{\partial \mathbf{h}[\mathbf{x}(k)]}{\partial \mathbf{x}(k)} \right _{\mathbf{x}(k) = \hat{\mathbf{x}}(k k-1)}$
$\mathbf{G}(k) = \mathbf{P}_e(k k-1)\mathbf{C}(k) [\mathbf{C}^T(k)\mathbf{P}_e(k k-1)\mathbf{C}(k) + \mathbf{P}_{n_1}(k)]^{-1}$
$\hat{\mathbf{x}}(k k) = \hat{\mathbf{x}}(k k-1) + \mathbf{G}(k) \{ \mathbf{y}(k) - \mathbf{h}[\hat{\mathbf{x}}(k k-1)] \}$
$\mathbf{P}_e(k k) = [\mathbf{I} - \mathbf{G}(k)\mathbf{C}^T(k)] \mathbf{P}_e(k k-1)$

### 2.1.3 Filtro de Kalman de Conjunto

O Filtro de Kalman de Conjunto (EnKF – *Ensemble Kalman Filter*) tem o objetivo de remediar o mau desempenho do EKF em aplicações onde as não linearidades são severas ou quando o número de variáveis de estado é alto [4]. Para isso, é realizado um conjunto de simulações de Monte Carlo, sugerido tipicamente 200 por DINIZ [4] ou até mesmo 500 por SUN e XIN [9].

O algoritmo propaga e atualiza os elementos do conjunto em cada ciclo de filtragem e não propaga a matriz de covariância diretamente. A estimativa da variável de estado e a matriz de covariância<sup>1</sup> do erro são substituídas pela média e pela covariância amostral dos elementos do conjunto, respectivamente [9]. Primeiro, um conjunto de amostras é gerado independentemente por uma determinada função densidade de probabilidade (PDF). Para um conjunto com  $J$  elementos  $\hat{\mathbf{x}}_j(k|k-1)$ , com o ruído  $\mathbf{n}_j(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_n(k))$ , temos, então:

$$\hat{\mathbf{x}}_j(k|k-1) = \mathbf{f}[\hat{\mathbf{x}}(k-1|k-1)] + \mathbf{n}_j(k) \quad , \quad (2.18)$$

<sup>1</sup>Dado que  $\mathbb{E}[\mathbf{e}(k|k)] = \mathbf{0}$ , a matriz de correlação do erro se torna igual à sua matriz de covariância. Como esta é a denominação majoritária nos artigos levantados, será adotada esta nomenclatura para este e os próximos algoritmos.

$$\bar{\mathbf{x}}(k|k-1) = \frac{1}{J} \sum_{j=1}^J \hat{\mathbf{x}}_j(k|k-1) \quad (2.19)$$

e

$$\bar{\mathbf{P}}_e(k|k-1) = \frac{1}{J-1} \sum_{j=1}^J [\hat{\mathbf{x}}_j(k|k-1) - \bar{\mathbf{x}}(k|k-1)] [\hat{\mathbf{x}}_j(k|k-1) - \bar{\mathbf{x}}(k|k-1)]^T, \quad (2.20)$$

que forma o conjunto de previsão, com uma estimativa de média  $\bar{\mathbf{x}}(k|k-1)$  e covariância  $\bar{\mathbf{P}}_e(k|k-1)$ .

Quando uma nova medição  $\mathbf{y}(k)$  está disponível, o conjunto de previsão produzido é corrigido por uma regra de atualização linear para um novo conjunto, denominado conjunto de análise. A partir da medição  $\mathbf{y}(k)$  são adicionadas perturbações  $\mathbf{n}_{2,j}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_{n_1}(k))$ , gerando um conjunto de  $J$  medições  $\hat{\mathbf{y}}_j(k)$ :

$$\hat{\mathbf{y}}_j(k) = \mathbf{y}(k) + \mathbf{n}_{2,j}(k), \quad (2.21)$$

sendo o valor médio e a matriz de covariância determinística do conjunto dados, respectivamente, por:

$$\bar{\mathbf{y}}(k) = \frac{1}{J} \sum_{j=1}^J \hat{\mathbf{y}}_j(k) \simeq \mathbf{y}(k) \quad (2.22)$$

e

$$\bar{\mathbf{P}}_y(k) = \frac{1}{J-1} \sum_{j=1}^J [\hat{\mathbf{y}}_j(k) - \bar{\mathbf{y}}(k)] [\hat{\mathbf{y}}_j(k) - \bar{\mathbf{y}}(k)]^T \simeq \frac{1}{J-1} \sum_{j=1}^J \mathbf{n}_{2,j}(k) \mathbf{n}_{2,j}^T(k). \quad (2.23)$$

Com isso, é possível calcular o ganho de Kalman a partir de (2.11) substituindo  $\mathbf{P}_e(k|k-1)$  por  $\bar{\mathbf{P}}_e(k|k-1)$  e  $\mathbf{P}_{n_1}(k)$  por  $\bar{\mathbf{P}}_y(k)$  e, na sequência, obter o estado atualizado com inovação  $\hat{\mathbf{x}}(k|k)$  aplicando-se (2.17) com  $\bar{\mathbf{x}}(k|k-1)$  e  $\bar{\mathbf{y}}(k)$  nos lugares de  $\hat{\mathbf{x}}(k|k-1)$  e  $\mathbf{y}(k)$ , respectivamente. É possível calcular, também, o conjunto de  $\hat{\mathbf{x}}_j(k|k)$  com a mesma equação, utilizando em (2.17) os valores dos elementos  $\hat{\mathbf{x}}_j(k|k-1)$  e  $\hat{\mathbf{y}}_j(k)$  nos lugares dos valores médios. A matriz de covariância atualizada  $\mathbf{P}_e(k|k)$  pode ser calculada por (2.12), substituindo  $\mathbf{P}_e(k|k-1)$  por  $\bar{\mathbf{P}}_e(k|k-1)$ .

O algoritmo apresentado por DINIZ [4] considera que a medição não ocorre em todos os instantes  $k$ . Sendo assim, quando há uma nova medição, é aplicada uma iteração simples do EKF utilizando as matrizes de covariâncias estimadas por simulação, ou seja, fazendo a substituição de  $\mathbf{P}_e(k|k-1)$  por  $\bar{\mathbf{P}}_e(k|k-1)$  e  $\mathbf{P}_{n_1}(k)$  por  $\bar{\mathbf{P}}_y(k)$ . Já nos demais instantes é feita etapa de previsão e análise com base na última medição. O autor ressalta que esta é uma forma de poupar processamento, não sendo a única forma do algoritmo. Já a forma proposta por SUN e XIN [9] não faz esta distinção, realizando o cálculo de forma única conforme as equações

descritas anteriormente. No entanto, se for realizado de forma única, o cálculo de  $\mathbf{P}_e(k|k-1)$  não pode ser feito apenas com a covariância de  $\hat{\mathbf{x}}_j(k|k-1)$  simulados, mas sim contemplando as duas etapas de uma vez, sendo da forma em (2.24). Esta forma é descrita por TIPPETT *et al.* [10] como sendo mais sofisticada para estimar  $\mathbf{P}_e(k|k-1)$  em filtros de conjunto baseados em MSE. O algoritmo final pode ser visto na Tabela 2.3.

$$\mathbf{P}_e(k|k-1) = \mathbf{A}(k-1)\mathbf{P}_e(k-1|k-1)\mathbf{A}^T(k-1) + \bar{\mathbf{P}}_e(k|k-1) \quad (2.24)$$

Referente à inicialização, é proposto em [4] e [9] que as variáveis sejam inicializadas conforme estimativas de valores esperados, com destaque que, para o algoritmo de DINIZ [4], pode ser necessário inicializar  $\bar{\mathbf{P}}_y(1) = \varepsilon\mathbf{I}$ , sendo  $\varepsilon$  uma constante pequena.

## 2.2 Modelos de *Bearing-Only Tracking*

Os modelos de BOT são baseados em análise de movimento relativo entre o navio-observador e o potencial alvo detectado. Como a única fonte de informação do alvo é a marcação medida pelo sensor passivo, suas variáveis de posição e velocidade não podem ser estimadas com o observador em movimento retilíneo uniforme (MRU), o que levaria a infinitas soluções possíveis.

Sendo assim, os primeiros modelos foram elaborados com base em uma análise geométrica, assumindo premissas como alvo em MRU e a necessidade de manobras do navio-observador ([11] – [19]). Posteriormente, foram estudados modelos com heurísticas iterativas e otimizantes, com destaque para Filtro de Kalman ([5], [20], [7], [6], [21] – [26], [8], [9] e [27]), que será aprofundado nesta pesquisa, além de técnicas como Filtro de Partículas ([28] e [29]), que apresentam bons resultados, mas com tempo de processamento quase 60 vezes maior que o Filtro de Kalman de Conjunto [9], e Algoritmo Genético [30], com menor recorrência na literatura.

### 2.2.1 Modelos Clássicos de BOT

Os modelos clássicos são baseados em uma análise geométrica da direção da fonte de sinal detectado em diferentes instantes, considerando seu movimento relativo ao navio-observador, que deve efetuar ao menos uma manobra, sendo cada trecho chamado de “pernada”<sup>2</sup>. Os dados de entrada são as marcações ( $\beta$ ) do alvo, além das informações do observador, como velocidade ( $v_o$ ) e rumo. O princípio geral dos

---

<sup>2</sup>Uma pernada corresponde ao movimento com velocidade e rumo constantes por um determinado tempo [18], mas o conceito pode ser flexibilizado para apenas rumo constante, dado que alguns modelos permitem variação do módulo da velocidade (com aceleração linear).

Tabela 2.3: Algoritmo do Filtro de Kalman de Conjunto

---

**Filtro de Kalman de Conjunto (EnKF)**


---

$\hat{\mathbf{x}}(0|0)$  e  $\mathbf{P}_e(0|0)$  estimados

Para  $k > 0$ :

$$\begin{aligned}\bar{\mathbf{x}}(k|k-1) &= \mathbf{0}; \bar{\mathbf{y}}(k) = \mathbf{0} \\ \bar{\mathbf{P}}_e(k|k-1) &= \mathbf{0}; \bar{\mathbf{P}}_y(k) = \mathbf{0}\end{aligned}$$

Para  $j$  de 1 a  $J$ :

$$\begin{aligned}\text{Gera } \mathbf{n}_j(k) \text{ e } \mathbf{n}_{2,j}(k) \\ \hat{\mathbf{x}}_j(k|k-1) &= \mathbf{f}[\hat{\mathbf{x}}(k-1|k-1)] + \mathbf{n}_j(k) \\ \bar{\mathbf{x}}(k|k-1) &= \bar{\mathbf{x}}(k|k-1) + \hat{\mathbf{x}}_j(k|k-1) \\ \hat{\mathbf{y}}_j(k) &= \mathbf{y}(k) + \mathbf{n}_{2,j}(k) \\ \bar{\mathbf{y}}(k) &= \bar{\mathbf{y}}(k) + \hat{\mathbf{y}}_j(k)\end{aligned}$$

$$\begin{aligned}\bar{\mathbf{x}}(k|k-1) &= \frac{1}{J}\bar{\mathbf{x}}(k|k-1) \\ \bar{\mathbf{y}}(k) &= \frac{1}{J}\bar{\mathbf{y}}(k)\end{aligned}$$

Para  $j$  de 1 a  $J$ :

$$\begin{aligned}\bar{\mathbf{P}}_e(k|k-1) &= \bar{\mathbf{P}}_e(k|k-1) \\ &\quad + [\hat{\mathbf{x}}_j(k|k-1) - \bar{\mathbf{x}}(k|k-1)][\hat{\mathbf{x}}_j(k|k-1) - \bar{\mathbf{x}}(k|k-1)]^T \\ \bar{\mathbf{P}}_y(k) &= \bar{\mathbf{P}}_y(k) + [\hat{\mathbf{y}}_j(k) - \bar{\mathbf{y}}(k)][\hat{\mathbf{y}}_j(k) - \bar{\mathbf{y}}(k)]^T\end{aligned}$$

$$\begin{aligned}\bar{\mathbf{P}}_e(k|k-1) &= \frac{1}{J-1}\bar{\mathbf{P}}_e(k|k-1) \\ \mathbf{P}_e(k|k-1) &= \mathbf{A}(k-1)\mathbf{P}_e(k-1|k-1)\mathbf{A}^T(k-1) + \bar{\mathbf{P}}_e(k|k-1) \\ \bar{\mathbf{P}}_y(k) &= \frac{1}{J-1}\bar{\mathbf{P}}_y(k) \\ \mathbf{C}(k) &= \left. \frac{\partial \mathbf{h}[\mathbf{x}(k)]}{\partial \mathbf{x}(k)} \right|_{\mathbf{x}(k) = \bar{\mathbf{x}}(k|k-1)} \\ \mathbf{G}(k) &= \mathbf{P}_e(k|k-1)\mathbf{C}(k)[\mathbf{C}^T(k)\mathbf{P}_e(k|k-1)\mathbf{C}(k) + \bar{\mathbf{P}}_y(k)]^{-1} \\ \hat{\mathbf{x}}(k|k) &= \bar{\mathbf{x}}(k|k-1) + \mathbf{G}(k)\{\bar{\mathbf{y}}(k) - \mathbf{h}[\bar{\mathbf{x}}(k|k-1)]\} \\ \mathbf{P}_e(k|k) &= [\mathbf{I} - \mathbf{G}(k)\mathbf{C}^T(k)]\mathbf{P}_e(k|k-1)\end{aligned}$$


---

modelos é a comparação entre a variação do rumo do observador ( $\alpha_o$ ) em função da manobra e a variação observada na marcação do alvo. Na Figura 2.1 é apresentado um exemplo de tática com duas pernadas, onde o cálculo considera a variação do rumo do navio-observador e a diferença entre as marcações obtidas após a manobra e as marcações esperadas se a manobra não ocorresse.

Os primeiros modelos de BOT remontam à Segunda Guerra Mundial, publicados no Manual da Rosa de Manobras da Marinha dos Estados Unidos (*Maneuvering Board Manual*, 1941) [11], onde eram necessárias 6 marcações e 2 pernadas, sendo 3 marcações antes da manobra e 3 depois, fazendo-se um cálculo geométrico manual por meio de desenho na rosa de manobras para estimar a velocidade do alvo ( $v_t$ ), o rumo ( $\alpha_t$ ) e a distância ( $r$  - range) do observador ao alvo.

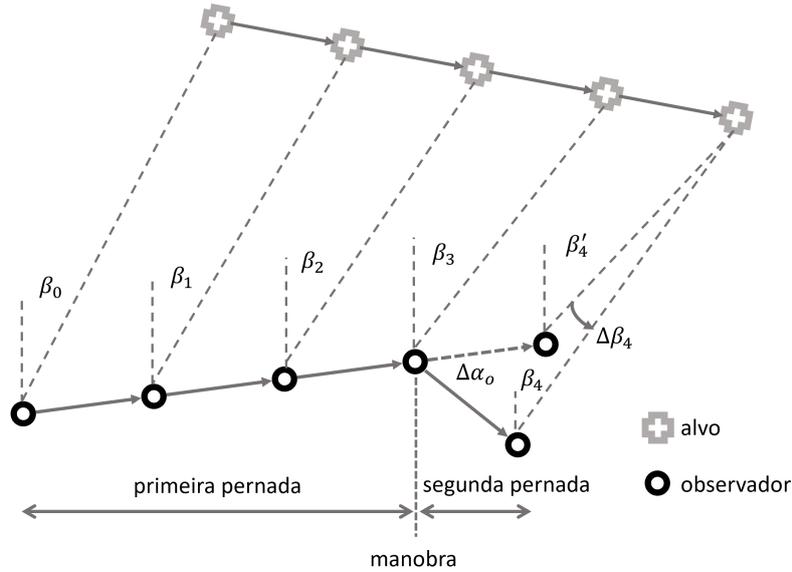


Figura 2.1: Exemplo de tática com duas pernadas

O método proposto por SPIESS [12], em 1953, consiste numa abordagem numérica do modelo geométrico, onde são necessárias 2 pernadas com velocidades distintas e 4 marcações, sendo 3 na primeira pernada e 1 na segunda. Já a técnica de EKELUND [13], de 1958, amplamente referenciada ([14] – [17]), também trabalha com 2 pernadas com velocidades distintas e 4 marcações, com 2 marcações antes da manobra e próximas entre si (sendo 1 imediatamente antes) e 2 após a manobra e também próximas entre si (sendo 1 imediatamente depois).

O trabalho de DOBBIE [14] (1971) avaliou a inclusão da terceira pernada e 6 marcações ao total (2 em cada pernada) e, com isso, obteve resultados mais precisos e que permitiram detectar erros quando o alvo mudasse de rumo ou velocidade. Para isso, trouxe uma abordagem sobre as fontes de erro dos modelos de BOT, que geralmente se concentram em quatro vieses: falta de acurácia das fórmulas, vieses nas observações das marcações do radar/sonar, viés estocástico e movimento não-linear ou não-uniforme do alvo. Referente ao último caso, os modelos levantados pressupõem um alvo em MRU, o que é uma aproximação do cenário de um navio se deslocando sem estar em manobra de combate. A não-linearidade do deslocamento traria um erro ao cálculo do modelo e o autor analisou a propagação destes erros para os estimadores dos parâmetros do alvo.

Os trabalhos posteriores e mais atuais com esta abordagem, como os de COLL [15] (1994), SON *et al.* [18] (2007) e LIN *et al.* [19] (2018), concluem que a manobra deve conter ao mínimo duas pernadas com velocidades e rumos diferentes, caso contrário acarretaria um número elevado de soluções possíveis. Por outro lado, um número excessivo de movimentos pode comprometer uma eventual ação ofensiva e, para o caso particular de um submarino (com mais restrições operativas), pode

impedir que se aproxime e se posicione dentro da zona de alcance do torpedo. Além disso, em geral, a mudança de rumo do observador na próxima pernada deve ser superior a  $50^\circ$  em relação à anterior [18].

Alguns trabalhos visaram estudar a manobra em si. BAKOS [16] (1995) estudou a melhor estratégia de pernadas para submarinos com propulsão diesel-elétrica. Já o trabalho de PASSERIEUX e VAN CAPPEL [31] (1998) buscou definir uma manobra ótima com base na matriz de informação de Fisher, usando critérios de acurácia global ou priorizando a acurácia da distância.

## 2.2.2 Modelos de BOT baseados em Filtro de Kalman

Os modelos de BOT baseados em Filtro de Kalman surgiram na década de 70, sendo as primeiras referências encontradas nos trabalhos de AIDALA (1976 [5] e 1979 [6]) e de LINGREN e GONG [20] (1978). Os estudos avaliaram o comportamento da aplicação do EKF, sendo concluído em [6] que a aplicação direta pode levar a um colapso prematuro da matriz de covariância do erro de estimação, impedindo a convergência do modelo. Para contornar esta dificuldade, propuseram uma alternativa que foi chamada de Filtro de Kalman Pseudolinear (PLKF – *Pseudo-Linear Kalman Filter*), com alteração no cálculo das matrizes de correlação e estabelecimento de inicialização adequada aos casos práticos sem comprometer significativamente o desempenho do modelo.

O EKF foi explorado por alguns autores em sua forma cartesiana (como por KOURKOULIS [22] em 1984, sendo mais usado em outros trabalhos apenas como comparativo), mas foi identificado por HOELZER *et al.* [7] (1978) e AIDALA e HAMMEL [21] (1983) que o uso de coordenadas polares descorrelacionava as componentes observáveis das não-observáveis do vetor de estado, prevenindo falsos valores de distância e colapso prematuro da matriz de correlação do erro de estimação por mal condicionamento, o que não era possível garantir no PLKF, sem a necessidade de impor restrição adicional na dinâmica do observador quando comparado ao modelo cartesiano. Neste modelo, embora a equação de saída seja linear, é necessário linearizar as equações de estado, sendo, então, conhecido por Filtro de Kalman Estendido com Coordenada Polar Modificada (MP-EKF – *Modified Polar Coordinate Extended Kalman Filter*).

O MP-EKF teve algumas variações exploradas: AIDALA e HAMMEL [21] (1983) e JAWAHAR e KOTESWARA RAO [26] (2016) trouxeram uma versão do algoritmo com unificação das etapas de transição de estados do algoritmo original; PEACH [23] (1997) aplicou o MP-EKF com uma inicialização baseada em parametrização de distância, onde um conjunto de filtros é executado em paralelo, sendo cada um inicializado com uma distância distinta e o resultado uma média dos resultados individu-

ais ponderada com o erro entre marcação medida e estimada (RP-MP-EKF – *Range Parameterized Modified Polar Coordinate Extended Kalman Filter*); e BREHARD e CADRE [24] (2006) e LA SCALA e MORELANDE [25] (2008) adotaram coordenada log-polar modificada (LP-EKF – *Log-Polar Coordinate Extended Kalman Filter*), onde houve indícios de maior robustez comparado à versão anterior, mas sendo a principal vantagem a simplificação do cálculo do Limite Inferior de Cramér-Rao (CRLB – *Cramér-Rao Lower Bound*). LA SCALA e MORELANDE [25] aplicaram a parametrização de distância ao LP-EKF (RP-LP-EKF), incluindo também a parametrização do módulo da velocidade inicial do alvo.

Já o PLKF foi revisto posteriormente por NGUYEN e DOĞANÇAY [8] (2017), que analisaram os vieses do PLKF em alvo com velocidade aproximadamente constante, propondo um algoritmo com termo de compensação destes (BC-PLKF – *Bias-Compensated Pseudo-Linear Kalman Filter*). Ainda assim, o viés decorrente da correlação do vetor de medição das marcações e o ruído de medição é reduzido com mais sucesso por meio do método proposto baseado em variáveis instrumentais, onde se busca decorrelacionar as variáveis no lugar de subtrair o viés como no BC-PLKF, e que deve ser aplicado de acordo com uma estratégia de seleção das medidas dos ângulos de marcação (SAM-IVKF – *Selective Angle Measurements Instrumental Variable-Based Kalman Filter*). O BC-PLKF ainda foi generalizado por KABA e TEMELTAS [27] (2022) para uso de ruído colorido (GBC-PLKF – *Generalized Bias-Compensated Pseudo-Linear Kalman Filter*).

A aplicação direta do EnKF não foi encontrada como um trabalho a parte. No entanto, seu conceito foi evoluído por SUN e XIN [9] (2020) no que foi chamado de Filtro de Kalman de Conjunto Aumentado (AEnKF – *Augmented Ensemble Kalman Filter*), onde o conjunto de medição é aumentado por uma expansão polinomial ortogonal e os resultados obtidos foram superiores aos demais algoritmos com pouco impacto no tempo de processamento.

Um resumo dos modelos pode ser visto na Figura 2.2. Esses algoritmos, com exceção do GBC-PLKF, serão detalhados no próximo capítulo. Além desses, também são encontradas algumas referências sobre o *Unscented Kalman Filter* [32], que também não será explorado neste trabalho em virtude de os resultados obtidos por este algoritmo em [9] se mostrarem inferiores aos obtidos nos demais modelos.

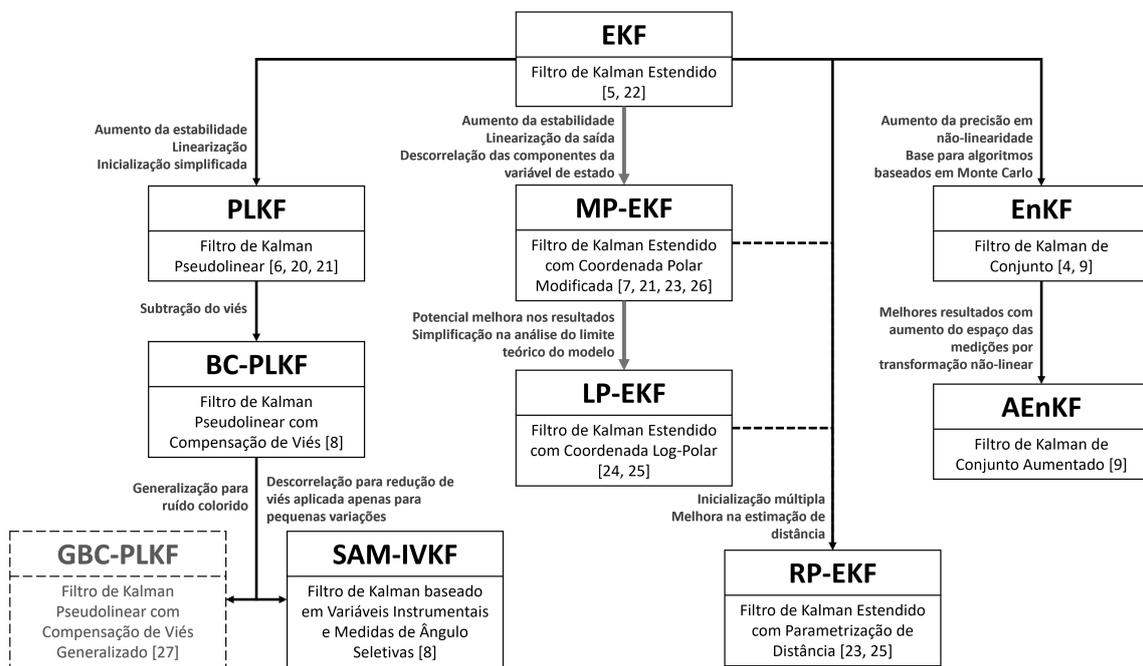


Figura 2.2: Principais algoritmos baseados em Filtro de Kalman

# Capítulo 3

## Aplicação de Filtro de Kalman ao Problema de *Bearing-Only Tracking*

Os Filtros de Kalman descritos na Seção 2.1 podem ser aplicados ao problema de BOT tanto na sua formulação direta quanto em versões alternativas, que buscam compensar vieses provocados pela linearização das equações ou mesmo melhorar a estabilidade do filtro. Para isso, primeiramente será definido um espaço de estados baseado na cinemática do movimento relativo de dois pontos materiais em duas dimensões, descrito na Seção 3.1.

Na sequência, os algoritmos da Figura 2.2, aplicados ao espaço de estados definido, serão detalhados nas demais seções deste capítulo, com exceção do GBC-PLKF, uma vez que este trabalho não entrará na discussão sobre as características dos ruídos e assumirá a premissa original do LKF de considerá-los GWN, assim como quase a totalidade dos modelos levantados. Particularmente para o MP-EKF e LP-EKF, o espaço de estados será convertido para as respectivas coordenadas, sem alterar as premissas de movimentação do alvo ou do observador.

### 3.1 Formulação do Espaço de Estados

O modelo consiste nas funções horárias de cinemática de dois objetos em duas dimensões no plano cartesiano ao longo do tempo, onde a orientação do eixo  $y$  representa o Norte, conforme Figura 3.1.

O alvo encontra-se em MRU, descrito pelo vetor posição do alvo  $\mathbf{s}_t(t) = [s_{t_x}(t) \ s_{t_y}(t)]^T$  e pelo seu vetor velocidade  $\mathbf{v}_t(t) = [v_{t_x} \ v_{t_y}]^T$  (aceleração  $\mathbf{a}_t(t) = \mathbf{0}$ , rumo constante  $\alpha_t$ ). O observador encontra-se em movimento que permite aceleração, sendo descrito pelos vetores posição  $\mathbf{s}_o(t) = [s_{o_x}(t) \ s_{o_y}(t)]^T$ , velocidade  $\mathbf{v}_o(t) = [v_{o_x}(t) \ v_{o_y}(t)]^T$  e aceleração  $\mathbf{a}_o(t) = [a_{o_x}(t) \ a_{o_y}(t)]^T$ , com rumo  $\alpha_o$ . A distância relativa é dada por  $\mathbf{r}(t) = [r_x(t) \ r_y(t)]^T$ , a velocidade relativa por

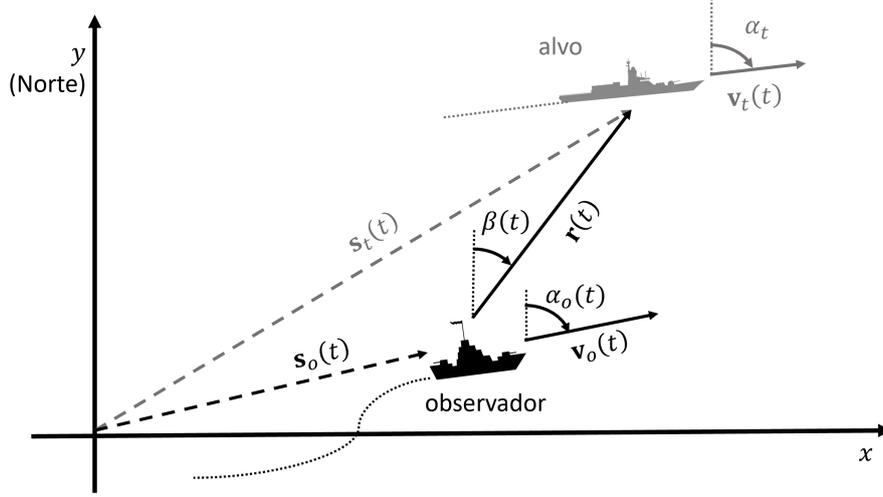


Figura 3.1: Representação da análise de BOT

$\mathbf{v}(t) = [v_x(t) \ v_y(t)]^T$  e a aceleração relativa por  $\mathbf{a}(t) = [a_x(t) \ a_y(t)]^T$ , sendo as equações de movimento relativo e suas relações dadas por:

$$\mathbf{a}(t) = \frac{\partial \mathbf{v}(t)}{\partial t} = \mathbf{a}_t(t) - \mathbf{a}_o(t) = -\mathbf{a}_o(t) \quad , \quad (3.1)$$

$$\begin{aligned} \mathbf{v}(t) &= \frac{\partial \mathbf{r}(t)}{\partial t} = \mathbf{v}_t(t) - \mathbf{v}_o(t) \\ &= \int \mathbf{a}(t) dt = \mathbf{v}(t_0) - \int_{t_0}^t \mathbf{a}_o(\xi) d\xi \end{aligned} \quad (3.2)$$

e

$$\begin{aligned} \mathbf{r}(t) &= \mathbf{s}_t(t) - \mathbf{s}_o(t) \\ &= \int \mathbf{v}(t) dt = \mathbf{r}(t_0) + \mathbf{v}(t_0)(t - t_0) - \int_{t_0}^t \xi \mathbf{a}_o(\xi) d\xi \quad . \end{aligned} \quad (3.3)$$

Discretizando no tempo com amostragem de período  $\tau$ , podemos definir como vetor de variáveis de estado:

$$\mathbf{x}(k) = [r_x(k) \ r_y(k) \ v_x(k) \ v_y(k)]^T = [\mathbf{r}^T(k) \ \mathbf{v}^T(k)]^T \quad . \quad (3.4)$$

São utilizadas como equações de estado as últimas equações de (3.3) e de (3.2). Fazendo  $t = k\tau$  e  $t_0 = (k-1)\tau$  e as mudanças adequadas de variáveis de integração, temos:

$$\begin{cases} \mathbf{r}(k) = \mathbf{r}(k-1) + \tau \mathbf{v}(k-1) - \int_0^\tau \xi \mathbf{a}_o(k\tau - \xi) d\xi \\ \mathbf{v}(k) = \mathbf{v}(k-1) - \int_0^\tau \mathbf{a}_o(k\tau - \xi) d\xi \quad . \end{cases} \quad (3.5)$$

Definindo a matriz de transição constante ao longo do tempo

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \tau & 0 \\ 0 & 1 & 0 & \tau \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \tau \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (3.6)$$

e o vetor auxiliar

$$\mathbf{w}(k) = \begin{bmatrix} \int_0^\tau \xi a_{o_x}(k\tau - \xi) d\xi \\ \int_0^\tau \xi a_{o_y}(k\tau - \xi) d\xi \\ \int_0^\tau a_{o_x}(k\tau - \xi) d\xi \\ \int_0^\tau a_{o_y}(k\tau - \xi) d\xi \end{bmatrix} = \begin{bmatrix} \int_0^\tau \xi \mathbf{a}_o(k\tau - \xi) d\xi \\ \int_0^\tau \mathbf{a}_o(k\tau - \xi) d\xi \end{bmatrix}, \quad (3.7)$$

podemos escrever a equação de transição de estado como:

$$\mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) - \mathbf{w}(k). \quad (3.8)$$

Os modelos propostos em [20], [8] e [9] simplificam o vetor  $\mathbf{w}(k)$ , suprimindo na primeira equação de (3.5) o termo integral relacionado à aceleração, como se a aceleração decorrente da manobra afetasse apenas a velocidade de forma impulsiva, fazendo com que a variação de posição fosse decorrente da velocidade já modificada em MRU. Além disso, na segunda equação de (3.5) é feita a simplificação da integral pela variação de velocidade do observador  $\Delta \mathbf{v}_o(k) = \mathbf{v}_o(k) - \mathbf{v}_o(k-1)$ . Esta segunda aproximação por aceleração média será importante para a aplicação prática, haja vista que a mensuração da aceleração instantânea do navio não é um dado extraído diretamente dos sistemas de combate, e, por isso, será empregada nos modelos que se seguem. Sendo assim, (3.7) é reescrita como:

$$\mathbf{w}(k) = \begin{bmatrix} \frac{\tau}{2} [v_{o_x}(k) - v_{o_x}(k-1)] \\ \frac{\tau}{2} [v_{o_y}(k) - v_{o_y}(k-1)] \\ v_{o_x}(k) - v_{o_x}(k-1) \\ v_{o_y}(k) - v_{o_y}(k-1) \end{bmatrix} = \begin{bmatrix} \frac{\tau}{2} [\mathbf{v}_o(k) - \mathbf{v}_o(k-1)] \\ \mathbf{v}_o(k) - \mathbf{v}_o(k-1) \end{bmatrix}. \quad (3.9)$$

A equação de saída, que relaciona a variável medida (marcação),  $\tilde{\beta}(k)$ , e as variáveis de estado, é uma relação não-linear dada por:

$$\tilde{\beta}(k) = h[\mathbf{x}(k)] + \eta(k) = \arctan \left[ \frac{r_x(k)}{r_y(k)} \right] + \eta(k), \quad (3.10)$$

onde  $\eta(k)$  é o ruído de medição definido como um processo aleatório gaussiano com média zero (GWN – *Gaussian White Noise*) e desvio-padrão  $\sigma_\eta$ , e  $\arctan$  a função arco tangente para 4 quadrantes [8]. O espaço de estados fica, então:

$$\begin{cases} \mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) - \mathbf{w}(k) \\ \tilde{\beta}(k) = \arctan \left[ \frac{r_x(k)}{r_y(k)} \right] + \eta(k) \end{cases}. \quad (3.11)$$

Em [22] são mencionadas quatro possíveis abordagens para considerar alvo em manobra ( $\mathbf{a}_t(t) \neq \mathbf{0}$ ), sendo três delas apenas uma visão conceitual (sem detalhamento algébrico) dada a dificuldade de aplicação prática. A quarta técnica se assemelha à proposta em [8], sendo, neste caso, a manobra do alvo correspondendo a apenas pequenas alterações de rumo. Para isso, a aceleração do alvo é considerada uma perturbação aleatória na equação de estado. Assim, o ruído de excitação  $\mathbf{n}(k)$  e sua matriz associada de geração das variáveis de estado  $\mathbf{B}(k)$  não são desprezados, existindo, então, a segunda parcela da soma em (2.4).

## 3.2 Aplicação Direta do Filtro de Kalman Estendido

O uso do EKF consiste na aplicação direta do espaço de estados definido anteriormente nas equações da Subseção 2.1.2. Neste caso,  $\mathbf{A}(k-1)$  já está definida em (3.6),  $\mathbf{B} = \mathbf{0}$ ,  $\mathbf{P}_{n_1}(k)$  é um escalar igual a  $\sigma_\eta^2(k)$  e, como  $h(\mathbf{x})$  é uma função escalar, (2.16) se torna um vetor e é dado por:

$$\begin{aligned} \mathbf{c}(k) &= \frac{\partial}{\partial \mathbf{x}(k)} \left\{ \arctan \left[ \frac{r_x(k)}{r_y(k)} \right] \right\}_{\mathbf{x}(k)=\hat{\mathbf{x}}(k|k-1)} \\ &= \begin{bmatrix} \frac{\hat{r}_y(k|k-1)}{\hat{r}_x^2(k|k-1) + \hat{r}_y^2(k|k-1)} & \frac{-\hat{r}_x(k|k-1)}{\hat{r}_x^2(k|k-1) + \hat{r}_y^2(k|k-1)} & 0 & 0 \end{bmatrix}^T \end{aligned} \quad (3.12)$$

Aplicando-se as demais equações, o algoritmo é resumido na Tabela 3.1.

Tabela 3.1: Filtro de Kalman Estendido aplicado a BOT

<b>EKF aplicado a BOT</b>
$\hat{\mathbf{x}}(0 0)$ e $\mathbf{P}_e(0 0)$ estimados
Para $k > 0$ :
$\mathbf{w}(k) = \left[ \frac{T}{2} [\mathbf{v}_o^T(k) - \mathbf{v}_o^T(k-1)] \quad [\mathbf{v}_o^T(k) - \mathbf{v}_o^T(k-1)] \right]^T$
$\hat{\mathbf{x}}(k k-1) = \mathbf{A}\hat{\mathbf{x}}(k-1 k-1) - \mathbf{w}(k)$
$\mathbf{P}_e(k k-1) = \mathbf{A}\mathbf{P}_e(k-1 k-1)\mathbf{A}^T$
$\mathbf{c}(k) = \begin{bmatrix} \frac{\hat{r}_y(k k-1)}{\hat{r}_x^2(k k-1) + \hat{r}_y^2(k k-1)} & \frac{-\hat{r}_x(k k-1)}{\hat{r}_x^2(k k-1) + \hat{r}_y^2(k k-1)} & 0 & 0 \end{bmatrix}^T$
$\mathbf{g}(k) = \mathbf{P}_e(k k-1)\mathbf{c}(k) [\mathbf{c}^T(k)\mathbf{P}_e(k k-1)\mathbf{c}(k) + \sigma_\eta^2(k)]^{-1}$
$\hat{\mathbf{x}}(k k) = \hat{\mathbf{x}}(k k-1) + \mathbf{g}(k) \left[ \tilde{\beta}(k) - \arctan \frac{\hat{r}_x(k k-1)}{\hat{r}_y(k k-1)} \right]$
$\mathbf{P}_e(k k) = [\mathbf{I} - \mathbf{g}(k)\mathbf{c}^T(k)] \mathbf{P}_e(k k-1)$

Como identificado posteriormente por AIDALA ([6] e [21]), este modelo depende

da inicialização para que haja convergência, não podendo ser adotada uma inicialização trivial.

KOURKOULIS [22] não entra na discussão sobre o assunto. LA SCALA e MORELANDE [25], JAWAHAR e KOTESWARA RAO [26] e SUN e XIN [9] propõem a inicialização com base em estimativa dos valores iniciais. Para isso, é necessário estimar a distância inicial  $\hat{r}(0)$ , a velocidade inicial do alvo  $\hat{v}_t(0)$  e o seu rumo  $\hat{\alpha}_t$ , assim como as respectivas variâncias  $\sigma_{\hat{r}}^2$ ,  $\sigma_{\hat{v}_t}^2$  e  $\sigma_{\hat{\alpha}_t}^2$ . Sendo assim, utilizando a marcação inicial  $\tilde{\beta}(0)$ , medida, e a velocidade inicial do observador  $v_{o_x}(0)$  e  $v_{o_y}(0)$ , conhecidas, são propostos  $\hat{\mathbf{x}}(0|0)$  e  $\mathbf{P}_e(0|0)$  da forma descrita em (3.13) e (3.14), respectivamente. Este método pode ser usado para os demais modelos baseados no EKF.

$$\hat{\mathbf{x}}(0|0) = \begin{bmatrix} \hat{r}(0) \sin \tilde{\beta}(0) & \hat{r}(0) \cos \tilde{\beta}(0) & \hat{v}_t(0) \sin \hat{\alpha}_t - v_{o_x}(0) & \hat{v}_t(0) \cos \hat{\alpha}_t - v_{o_y}(0) \end{bmatrix}^T \quad (3.13)$$

$$\mathbf{P}_e(0|0) = \begin{bmatrix} \mathbf{P}_e^{(r)}(0|0) & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_e^{(v)}(0|0) \end{bmatrix}$$

$$\mathbf{P}_e^{(r)}(0|0) = \begin{bmatrix} \hat{r}^2(0) \sigma_{\eta}^2 \cos^2 \tilde{\beta}(0) + \sigma_{\hat{r}}^2 \sin^2 \tilde{\beta}(0) & [\sigma_{\hat{r}}^2 - \hat{r}^2(0) \sigma_{\eta}^2] \sin \tilde{\beta}(0) \cos \tilde{\beta}(0) \\ [\sigma_{\hat{r}}^2 - \hat{r}^2(0) \sigma_{\eta}^2] \sin \tilde{\beta}(0) \cos \tilde{\beta}(0) & \hat{r}^2(0) \sigma_{\eta}^2 \sin^2 \tilde{\beta}(0) + \sigma_{\hat{r}}^2 \cos^2 \tilde{\beta}(0) \end{bmatrix}$$

$$\mathbf{P}_e^{(v)}(0|0) = \begin{bmatrix} \hat{v}_t^2(0) \sigma_{\hat{\alpha}_t}^2 \cos^2 \hat{\alpha}_t + \sigma_{\hat{v}_t}^2 \sin^2 \hat{\alpha}_t & [\sigma_{\hat{v}_t}^2 - \hat{v}_t^2(0) \sigma_{\hat{\alpha}_t}^2] \sin \hat{\alpha}_t \cos \hat{\alpha}_t \\ [\sigma_{\hat{v}_t}^2 - \hat{v}_t^2(0) \sigma_{\hat{\alpha}_t}^2] \sin \hat{\alpha}_t \cos \hat{\alpha}_t & \hat{v}_t^2(0) \sigma_{\hat{\alpha}_t}^2 \sin^2 \hat{\alpha}_t + \sigma_{\hat{v}_t}^2 \cos^2 \hat{\alpha}_t \end{bmatrix} \quad (3.14)$$

### 3.3 Filtro de Kalman Pseudolinear

Para evitar alguns problemas de convergência do EKF, LINGREN e GONG [20] e AIDALA [6] propuseram adaptações na linearização do espaço de estado, assim como na inicialização do algoritmo.

A primeira mudança consiste na manipulação algébrica de (3.10), onde a partir da definição da tangente da marcação  $\beta$  e que  $\tilde{\beta}(k) = \beta(k) + \eta(k)$ , obtém-se as relações

$$0 = \hat{\mathbf{c}}(k) \mathbf{x}(k) + \nu(k) \quad (3.15)$$

$$\nu(k) = r(k) \sin \eta(k) \quad (3.16)$$

$$\hat{\mathbf{c}}(k) = \begin{bmatrix} \cos \tilde{\beta}(k) & -\sin \tilde{\beta}(k) & 0 & 0 \end{bmatrix}^T \quad (3.17)$$

onde  $\hat{\mathbf{c}}(k)$  passa a ser dependente da variável de medição (e, portanto, do seu ruído), e não uma função da variável de estado, como na linearização do EKF. Esta correlação entre  $\hat{\mathbf{c}}(k)$  e  $\eta(k)$  se torna uma fonte de viés, sendo discutido posteriormente em [8]

e apresentado na Seção 3.4.

Em [6] é demonstrado que o ruído pseudolinear  $\nu(k)$  tem valor esperado igual a 0 e que  $\mathbb{E}[\nu^2(k)] \approx r^2(k)\sigma_\eta^2(k)$  para  $\sigma_\eta(k) \ll 1$ . Quando aplicado ao EKF, é proposta a simplificação de utilizar apenas a variância do erro da medição na equação do ganho (2.11) dado que, numa aplicação operacional realista, o valor de  $r(k)$  é desconhecido, fazendo, então,  $\mathbf{P}_{n_1}(k)$  ser um escalar igual a  $\sigma_\eta^2(k)$ , como no EKF. É demonstrado ainda que, embora simplória, essa simplificação descorrelaciona efetivamente a estimativa de  $\mathbf{P}_{n_1}(k)$  da distância ao alvo  $r(k)$  sem distorcer seriamente as características de ponderação estatística do filtro.

Sendo assim, as mudanças propostas impactam em (2.11), (2.12) e (3.12) ao substituir  $\mathbf{c}(k)$  por  $\hat{\mathbf{c}}(k)$ , além de modificar (2.17) por:

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) - \mathbf{g}(k)\hat{\mathbf{c}}^T(k)\hat{\mathbf{x}}(k|k-1) . \quad (3.18)$$

Outra discussão é feita sobre a inicialização do modelo. Em particular, uma vez que não existe uma solução única de BOT antes da primeira manobra do próprio navio, pode ser aconselhável tomar precauções durante este período para evitar que a estimativa da solução inicial seja mal interpretada como sendo precisa e que produza um histórico de resultados enviesados antes da manobra. Com isso, é proposta a inicialização trivial ( $\hat{\mathbf{x}}(0|0) = \mathbf{0}$  e  $\mathbf{P}_e(0|0) = \mathbf{I}$ ) dado que não é viável obter de forma prática um valor esperado destas variáveis. Esta inicialização também garante que a matriz de correlação não colapse, uma vez que a solução fica travada no valor inicial até a manobra do navio-observador, o que não pode ser garantido com uma inicialização aleatória.

Com esta inicialização e alterações propostas, o algoritmo PLKF pode ser definido com base no EKF, conforme Tabela 3.2.

Tabela 3.2: Filtro de Kalman Pseudolinear

<b>PLKF</b>
$\hat{\mathbf{x}}(0 0) = \mathbf{0}$ $\mathbf{P}_e(0 0) = \mathbf{I}$
Para $k > 0$ : $\mathbf{w}(k) = \left[ \frac{T}{2} [\mathbf{v}_o^T(k) - \mathbf{v}_o^T(k-1)] \quad [\mathbf{v}_o^T(k) - \mathbf{v}_o^T(k-1)] \right]^T$ $\hat{\mathbf{x}}(k k-1) = \mathbf{A}\hat{\mathbf{x}}(k-1 k-1) - \mathbf{w}(k)$ $\mathbf{P}_e(k k-1) = \mathbf{A}\mathbf{P}_e(k-1 k-1)\mathbf{A}^T$ $\hat{\mathbf{c}}(k) = \left[ \cos \tilde{\beta}(k) \quad -\sin \tilde{\beta}(k) \quad 0 \quad 0 \right]^T$ $\mathbf{g}(k) = \mathbf{P}_e(k k-1)\hat{\mathbf{c}}(k) [\hat{\mathbf{c}}^T(k)\mathbf{P}_e(k k-1)\hat{\mathbf{c}}(k) + \sigma_\eta^2(k)]^{-1}$ $\hat{\mathbf{x}}(k k) = \hat{\mathbf{x}}(k k-1) - \mathbf{g}(k)\hat{\mathbf{c}}^T(k)\hat{\mathbf{x}}(k k-1)$ $\mathbf{P}_e(k k) = [\mathbf{I} - \mathbf{g}(k)\hat{\mathbf{c}}^T(k)] \mathbf{P}_e(k k-1)$

### 3.4 Filtro de Kalman Pseudolinear com Compensação de Viés

O BC-PLKF, estabelecido em [8], parte do mesmo princípio do PLKF para redefinir a relação entre os estados e a saída vista em (3.15) e (3.17)<sup>1</sup>. A primeira mudança adotada foi a possibilidade de considerar o alvo com pequena variação de rota, sendo a manobra do alvo modelada como uma perturbação aleatória independente GWN definida como  $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ . Desta forma, a equação de transição de estados (3.8) é reescrita como

$$\mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) + \mathbf{u}(k) - \mathbf{w}(k) , \quad (3.19)$$

onde

$$\mathbf{Q} = \begin{bmatrix} q_x \frac{\tau^3}{3} & 0 & q_x \frac{\tau^2}{2} & 0 \\ 0 & q_y \frac{\tau^3}{3} & 0 & q_y \frac{\tau^2}{2} \\ q_x \frac{\tau^2}{2} & 0 & q_x \tau & 0 \\ 0 & q_y \frac{\tau^2}{2} & 0 & q_y \tau \end{bmatrix} \quad (3.20)$$

e  $q_x$  e  $q_y$  são as componentes  $x$  e  $y$  da densidade espectral de potência do processo aleatório  $\mathbf{u}(k)$ , dadas na grandeza *distância*<sup>2</sup>/*tempo*<sup>3</sup> e assumidos constantes ao longo do tempo. Esta mudança impacta na equação de  $\mathbf{P}_e(k|k-1)$  (2.4), com  $\mathbf{B}(k) = \mathbf{I}$  e  $\mathbf{P}_n(k) = \mathbf{Q}$ , sendo reescrita como:

$$\mathbf{P}_e(k|k-1) = \mathbf{A}\mathbf{P}_e(k-1|k-1)\mathbf{A}^T + \mathbf{Q} . \quad (3.21)$$

Fazendo-se  $q_x = q_y = 0$ , teremos  $\mathbf{Q} = \mathbf{0}$  e valerá a premissa de MRU do alvo como definida no EKF e no PLKF.

Outra diferença em relação ao PLKF foi a não simplificação da variância do ruído pseudolinear  $\nu(k)$  adotada por AIDALA [6]. Sendo assim, no lugar de desconsiderar o termo  $r(k)$ , desconhecido, estima-se o mesmo pela variável de estado  $\hat{\mathbf{x}}(k|k-1)$ , sendo então  $\mathbf{P}_{n_1}(k)$  um escalar  $P_\nu(k)$  dado por:

$$P_\nu(k) = [\hat{r}_x^2(k|k-1) + \hat{r}_y^2(k|k-1)] \sigma_\eta^2(k) , \quad (3.22)$$

sendo essa relação aplicada na equação do ganho (2.11).

Como mencionada na Seção 3.3, uma das fontes de viés do PLKF discutida por

---

<sup>1</sup>O espaço de estados apresentado em [8] considera apenas as variáveis de posição e velocidade do alvo (e não o movimento relativo), além de usar o ângulo da marcação na convenção trigonométrica (e não na náutica). Sendo assim, as equações foram reescritas para que a notação fique em conformidade com os demais artigos e modelos estudados.

NGUYEN e DOĞANÇAY [8] é a utilização do valor de medição da marcação  $\tilde{\beta}$  (carregando seu ruído) para definir  $\hat{\mathbf{c}}(k)$ , que em um modelo tradicional linear seria um vetor (ou matriz) determinístico(a), sem correlação com o ruído de medição  $\eta(k)$ . Outra fonte de viés, que se torna significativa em situações com alvo em manobras rápidas, é a dependência de  $\hat{\mathbf{c}}(k)$  em relação a  $\mathbf{x}(k)$ , dado que a marcação depende da posição relativa do alvo contida na variável de estado.

A determinação do viés é feita por meio da subtração entre  $\hat{\mathbf{x}}(k|k)$  (3.18) e  $\mathbf{x}(k)$  (3.19), onde na primeira equação é feita uma manipulação algébrica com a substituição do valor do ganho de Kalman (2.11), aplicado o lema da inversão de matriz e, posteriormente, aplicadas as equações de  $\hat{\mathbf{x}}(k|k-1)$  (2.5) e a equação manipulada de saída do PLKF (3.15). Assim, é possível obter a relação a seguir que relaciona o estado previsto com inovação e o estado real atuais ( $\hat{\mathbf{x}}(k|k) - \mathbf{x}(k)$ ), o estado previsto com inovação e o estado real anteriores ( $\hat{\mathbf{x}}(k-1|k-1) - \mathbf{x}(k-1)$ ), os termos referentes à aceleração que provoca a variação de rumo dos navios alvo e observador ( $\mathbf{u}(k) - \mathbf{w}(k)$ ) e o erro de medição pseudolinear  $\nu(k)$ :

$$\begin{aligned} \hat{\mathbf{x}}(k|k) - \mathbf{x}(k) = & \left\{ \mathbf{I} - [\mathbf{P}_e^{-1}(k|k-1) + \hat{\mathbf{c}}(k)P_\nu^{-1}(k)\hat{\mathbf{c}}^T(k)]^{-1} \hat{\mathbf{c}}(k)P_\nu^{-1}(k)\hat{\mathbf{c}}^T(k) \right\} \\ & \times \{ \mathbf{A} [\hat{\mathbf{x}}(k-1|k-1) - \mathbf{x}(k-1)] - [\mathbf{u}(k) - \mathbf{w}(k)] \} \\ & + [\mathbf{P}_e^{-1}(k|k-1) + \hat{\mathbf{c}}(k)P_\nu^{-1}(k)\hat{\mathbf{c}}^T(k)]^{-1} \hat{\mathbf{c}}(k)P_\nu^{-1}(k)\nu(k) \end{aligned} \quad (3.23)$$

Aplicando-se novamente o lema da inversão de matriz no primeiro termo entre chaves de (3.23), é possível escrever a relação em 3 parcelas:

$$\hat{\mathbf{x}}(k|k) - \mathbf{x}(k) = \gamma_1(k) + \gamma_2(k) + \gamma_3(k) \quad (3.24)$$

sendo

$$\gamma_1(k) = [\mathbf{P}_e^{-1}(k|k-1) + \hat{\mathbf{c}}(k)P_\nu^{-1}(k)\hat{\mathbf{c}}^T(k)]^{-1} \mathbf{P}_e^{-1}(k|k-1)\mathbf{A} [\hat{\mathbf{x}}(k-1|k-1) - \mathbf{x}(k-1)] , \quad (3.25)$$

$$\gamma_2(k) = - [\mathbf{P}_e^{-1}(k|k-1) + \hat{\mathbf{c}}(k)P_\nu^{-1}(k)\hat{\mathbf{c}}^T(k)]^{-1} \mathbf{P}_e^{-1}(k|k-1) [\mathbf{u}(k) - \mathbf{w}(k)] \quad (3.26)$$

e

$$\gamma_3(k) = [\mathbf{P}_e^{-1}(k|k-1) + \hat{\mathbf{c}}(k)P_\nu^{-1}(k)\hat{\mathbf{c}}^T(k)]^{-1} \hat{\mathbf{c}}(k)P_\nu^{-1}(k)\nu(k) . \quad (3.27)$$

O viés  $\epsilon(k)$  é dado pelo valor esperado de (3.24):

$$\epsilon(k) = \mathbb{E} [\hat{\mathbf{x}}(k|k) - \mathbf{x}(k)] = \mathbb{E} [\gamma_1(k)] + \mathbb{E} [\gamma_2(k)] + \mathbb{E} [\gamma_3(k)] . \quad (3.28)$$

A análise de cada parcela de (3.28) é interpretada como:

i.  $\mathbb{E} [\gamma_1(k)]$  é o termo que propaga o viés do instante  $k-1$  para  $k$ , ou seja, se

forem implementadas condições para que  $\mathbb{E}[\gamma_2(k)] = \mathbb{E}[\gamma_3(k)] = \mathbf{0}$ , então teríamos  $\mathbb{E}[\gamma_1(k)] = \mathbf{0}$ ;

- ii.  $\mathbb{E}[\gamma_2(k)]$  depende da correlação entre  $\hat{\mathbf{c}}(k)$  e  $\mathbf{u}(k)$  (lembrando que  $\mathbf{w}(k)$  depende apenas da variação de velocidade do observador, o que é conhecido e determinístico). Dado que  $\hat{\mathbf{c}}(k)$  depende de  $\tilde{\beta}(k)$ , que, embora medido, pode ser obtido por uma função da posição relativa do alvo, contida na variável de estado  $\mathbf{x}(k)$  (3.19), e que esta variável é função de uma parcela de  $\mathbf{u}(k)$ , quanto menor a influência de  $\mathbf{u}(k)$  em  $\mathbf{x}(k)$ , menor a correlação. Por isso, em [8] é assumido que, para alvo com pequenas variações na trajetória ( $q_x$  e  $q_y$  pequenos), considera-se que  $\mathbb{E}[\gamma_2(k)] \approx \mathbf{0}$ ; e
- iii.  $\mathbb{E}[\gamma_3(k)]$  é o termo mais relevante para o viés do modelo, já que a correlação entre  $\hat{\mathbf{c}}(k)$  e  $\nu(k)$  é forte, pois o primeiro é função de  $\tilde{\beta}(k)$ , que contém o ruído de medição  $\eta(k)$ , e o segundo é uma função direta deste ruído.

Tendo em vista esse viés, é proposto que a cada iteração do algoritmo seja estimado qual o valor instantâneo de  $\gamma_3(k)$  e que seja subtraído da variável de estado estimada a fim de se compensar o efeito do viés. Como o valor instantâneo de  $\hat{\mathbf{c}}(k)\nu(k)$  não pode ser determinado, é proposta uma aproximação pelo valor esperado condicional  $\mathbb{E}[\hat{\mathbf{c}}(k)\nu(k)|\mathbf{x}(k)] = -\sigma_\eta^2(k)\mathbf{M}\mathbf{x}(k)$  e, então, aplicado o valor de  $\hat{\mathbf{x}}(k|k)$  para estimar  $r(k)$ :

$$\hat{\gamma}_3(k) = - [\mathbf{P}_e^{-1}(k|k-1) + \hat{\mathbf{c}}(k)P_\nu^{-1}(k)\hat{\mathbf{c}}^T(k)]^{-1} P_\nu^{-1}(k)\sigma_\eta^2(k)\mathbf{M}\hat{\mathbf{x}}(k|k) \quad , \quad (3.29)$$

onde:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad . \quad (3.30)$$

Ainda é possível simplificar (3.29) usando  $\mathbf{P}_e(k|k)$  já calculado no lugar de  $[\mathbf{P}_e^{-1}(k|k-1) + \hat{\mathbf{c}}(k)P_\nu^{-1}(k)\hat{\mathbf{c}}^T(k)]^{-1}$  (verificável pelo lema de inversão de matriz). Com isso, o algoritmo se assemelha ao da Tabela 3.2, com a inclusão da etapa de subtração do viés (3.29) de  $\hat{\mathbf{x}}(k|k)$  após seu cálculo:

$$\begin{aligned} \hat{\mathbf{x}}_{BC}(k|k) &= \hat{\mathbf{x}}(k|k) - \hat{\gamma}_3(k) \\ &= \hat{\mathbf{x}}(k|k) + \mathbf{P}_e(k|k)P_\nu^{-1}(k)\sigma_\eta^2(k)\mathbf{M}\hat{\mathbf{x}}(k|k) \end{aligned} \quad (3.31)$$

e  $\hat{\mathbf{x}}_{BC}(k|k)$  é utilizado para o próximo *loop* do algoritmo.

Os autores de [8] ressaltam que, pela definição da covariância do erro (2.10) e devido ao viés (3.28), o valor de  $\mathbb{E}\left\{[\hat{\mathbf{x}}(k|k) - \mathbf{x}(k)][\hat{\mathbf{x}}(k|k) - \mathbf{x}(k)]^T\right\}$

seria maior que  $\mathbf{P}_e(k|k)$ ; mas, considerando  $\boldsymbol{\epsilon}(k) \approx \hat{\boldsymbol{\gamma}}_3(k)$ , tem-se que  $\mathbb{E} \left\{ [\hat{\mathbf{x}}_{BC}(k|k) - \mathbf{x}(k)] [\hat{\mathbf{x}}_{BC}(k|k) - \mathbf{x}(k)]^T \right\} \approx \mathbf{P}_e(k|k)$ , não sendo necessário alterar (2.10).

No entanto, não é feita uma discussão sobre a inicialização do modelo, sendo indicado o uso do valor esperado das variáveis no instante inicial, embora não seja uma informação disponível em uma situação real. Sendo assim, o algoritmo pode ser resumido na Tabela 3.3.

Tabela 3.3: Filtro de Kalman Pseudolinear com Compensação de Viés

<b>BC-PLKF</b>
$\hat{\mathbf{x}}_{BC}(0 0)$ e $\mathbf{P}_e(0 0)$ estimados
Para $k > 0$ :
$\mathbf{w}(k) = \left[ \frac{\tau}{2} [\mathbf{v}_o^T(k) - \mathbf{v}_o^T(k-1)] \quad [\mathbf{v}_o^T(k) - \mathbf{v}_o^T(k-1)] \right]^T$
$\hat{\mathbf{x}}(k k-1) = \mathbf{A}\hat{\mathbf{x}}_{BC}(k-1 k-1) - \mathbf{w}(k)$
$\mathbf{P}_e(k k-1) = \mathbf{A}\mathbf{P}_e(k-1 k-1)\mathbf{A}^T + \mathbf{Q}$
$\hat{\mathbf{c}}(k) = \left[ \cos \tilde{\beta}(k) \quad -\sin \tilde{\beta}(k) \quad 0 \quad 0 \right]^T$
$P_\nu(k) = [\hat{r}_x^2(k k-1) + \hat{r}_y^2(k k-1)] \sigma_\eta^2(k)$
$\mathbf{g}(k) = \mathbf{P}_e(k k-1)\hat{\mathbf{c}}(k) [\hat{\mathbf{c}}^T(k)\mathbf{P}_e(k k-1)\hat{\mathbf{c}}(k) + P_\nu(k)]^{-1}$
$\hat{\mathbf{x}}(k k) = \hat{\mathbf{x}}(k k-1) - \mathbf{g}(k)\hat{\mathbf{c}}^T(k)\hat{\mathbf{x}}(k k-1)$
$\mathbf{P}_e(k k) = [\mathbf{I} - \mathbf{g}(k)\hat{\mathbf{c}}^T(k)] \mathbf{P}_e(k k-1)$
$\hat{\mathbf{x}}_{BC}(k k) = \hat{\mathbf{x}}(k k) + \mathbf{P}_e(k k)P_\nu^{-1}(k)\sigma_\eta^2(k)\mathbf{M}\hat{\mathbf{x}}(k k)$

### 3.5 Filtro de Kalman baseado em Variáveis Instrumentais e Medidas de Ângulo Seletivas

As variáveis instrumentais (*Instrumental Variables – IV*) são uma técnica de estimação que fornece uma maneira de obter estimativas de parâmetros consistentes. Este método é amplamente utilizado em econometria, com poucas aplicações fora da área. Esta técnica pode ser vista como uma generalização da regressão linear por Mínimos Quadrados Ordinários porque, enquanto as estimativas da regressão linear medem apenas a magnitude da associação das variáveis explicativas e da variável dependente, as variáveis instrumentais permitem avaliar também a direção de causalidade [33]. De forma resumida, uma variável instrumental é aquela correlacionada à variável explicativa, mas sendo correlacionada à variável dependente somente por meio da variável explicativa, ou seja, sem ser correlacionada ao resíduo (parte não explicada) do modelo.

Tendo o objetivo de retirar o viés do modelo PLKF (3.28) decorrelacionando  $\hat{\mathbf{c}}(k)$  do ruído de medição de  $\tilde{\beta}(k)$ ,  $\eta(k)$  (e não mais fazendo uma subtração do viés

como no BC-PLKF), NGUYEN e DOĞANÇAY [8] buscaram estabelecer uma variável fortemente correlacionada à  $\hat{\mathbf{c}}(k)$ , mas sem correlação com o ruído do modelo. Como o método de IV foi desenvolvido originalmente para parâmetros determinísticos (em vez de aleatório), os autores fizeram a derivação do Filtro de Kalman baseado no RLS em paralelo com o método de IV, com a simplificação de  $\mathbf{A} = \mathbf{I}$  e  $\mathbf{Q} = \mathbf{0}$ , tornando a variável  $\mathbf{x}(k)$  determinística. Com isso, a única diferença encontrada foi com relação à equação do ganho, que passou a ser escrita como:

$$\mathbf{g}_{IV}(k) = \mathbf{P}_e(k|k-1)\hat{\mathbf{c}}_{IV}(k)[\hat{\mathbf{c}}^T(k)\mathbf{P}_e(k|k-1)\hat{\mathbf{c}}_{IV}(k) + P_\nu(k)]^{-1} . \quad (3.32)$$

Sendo  $\hat{\mathbf{c}}_{IV}(k)$  descorrelacionado de  $\eta(k)$ , aplicando a modificação no vetor de ganho do RLS em (3.27) faz com que  $\mathbb{E}[\gamma_3(k)] \approx \mathbf{0}$ . A definição do valor ótimo de  $\hat{\mathbf{c}}_{IV}(k)$  seria similar a (3.17), substituindo  $\tilde{\beta}(k)$  pelo valor verdadeiro sem ruído  $\beta(k)$ . Como este valor é desconhecido, é sugerido em [8] que se utilize um valor sub-ótimo estimando a marcação com a variável sem viés  $\hat{\mathbf{x}}_{BC}(k|k)$ , dada em (3.31). Logo:

$$\hat{\mathbf{c}}_{IV}(k) = \begin{bmatrix} \cos \hat{\beta}_{BC}(k|k) & -\sin \hat{\beta}_{BC}(k|k) & 0 & 0 \end{bmatrix}^T , \quad (3.33)$$

onde

$$\hat{\beta}_{BC}(k|k) = \arctan \frac{\hat{r}_{x,BC}(k|k)}{\hat{r}_{y,BC}(k|k)} , \quad (3.34)$$

relembrando que  $\hat{r}_{x,BC}(k|k)$  e  $\hat{r}_{y,BC}(k|k)$  são, respectivamente, a primeira e segunda componentes de  $\hat{\mathbf{x}}_{BC}(k|k)$ .

Esta aproximação torna o algoritmo dependente do BC-PLKF. Então, ao final de cada *loop* descrito na Tabela 3.3, deverá ser incluído o cálculo de  $\hat{\mathbf{c}}_{IV}(k)$  (3.33) para, então, calcular o novo ganho  $\mathbf{g}_{IV}(k)$  (3.32). Assim, as equações de  $\hat{\mathbf{x}}(k|k)$  (3.18) e  $\mathbf{P}_e(k|k)$  (2.12) podem ser recalculadas utilizando esse novo valor de ganho, sem alterar as demais variáveis.

Como o método de IV requer forte correlação entre a variável explicativa  $\hat{\mathbf{c}}(k)$  e a variável instrumental  $\hat{\mathbf{c}}_{IV}(k)$  para garantir a *performance* do MSE, em [8] é proposto que se use uma estratégia de seleção com base nos ângulos medido e calculado. Calculando  $\Delta\beta(k) = \left| \hat{\beta}_{BC}(k) - \tilde{\beta}(k) \right|$ , a correlação entre as variáveis é forte quando a diferença for pequena e, assim, recomenda-se aplicar esta etapa adicional de IV descrita no parágrafo anterior. Por outro lado, não aplicar essa etapa no algoritmo prejudica a redução do viés pois o algoritmo se resumiria ao BC-PLKF apenas. Dessa forma, visando ao equilíbrio entre boa performance do MSE e redução no viés, é sugerido um limite  $\Delta\beta(k) < \kappa\sigma_\eta(k)$  para aplicação das IV, com  $2 \leq \kappa \leq 4$  [34]. O algoritmo final consta na Tabela 3.4.

Tabela 3.4: Filtro de Kalman baseado em Variáveis Instrumentais e Medidas de Ângulo Seletivas

---

**SAM-IVKF**

---

$\hat{\mathbf{x}}_{IV}(0|0)$  e  $\mathbf{P}_{e,IV}(0|0)$  estimados

Para  $k > 0$ :

$$\begin{aligned} \mathbf{w}(k) &= \left[ \frac{\tau}{2} [\mathbf{v}_o^T(k) - \mathbf{v}_o^T(k-1)] \quad [\mathbf{v}_o^T(k) - \mathbf{v}_o^T(k-1)] \right]^T \\ \hat{\mathbf{x}}(k|k-1) &= \mathbf{A}\hat{\mathbf{x}}_{IV}(k-1|k-1) - \mathbf{w}(k) \\ \mathbf{P}_e(k|k-1) &= \mathbf{A}\mathbf{P}_{e,IV}(k-1|k-1)\mathbf{A}^T + \mathbf{Q} \\ \hat{\mathbf{c}}(k) &= \left[ \cos \tilde{\beta}(k) \quad -\sin \tilde{\beta}(k) \quad 0 \quad 0 \right]^T \\ P_\nu(k) &= [\hat{r}_x^2(k|k-1) + \hat{r}_y^2(k|k-1)] \sigma_\eta^2(k) \\ \mathbf{g}(k) &= \mathbf{P}_e(k|k-1)\hat{\mathbf{c}}(k) [\hat{\mathbf{c}}^T(k)\mathbf{P}_e(k|k-1)\hat{\mathbf{c}}(k) + P_\nu(k)]^{-1} \\ \hat{\mathbf{x}}(k|k) &= \hat{\mathbf{x}}(k|k-1) - \mathbf{g}(k)\hat{\mathbf{c}}^T(k)\hat{\mathbf{x}}(k|k-1) \\ \mathbf{P}_e(k|k) &= [\mathbf{I} - \mathbf{g}(k)\hat{\mathbf{c}}^T(k)] \mathbf{P}_e(k|k-1) \\ \hat{\mathbf{x}}_{BC}(k|k) &= \hat{\mathbf{x}}(k|k) + \mathbf{P}_e(k|k)P_\nu^{-1}(k)\sigma_\eta^2(k)\mathbf{M}\hat{\mathbf{x}}(k|k) \\ \hat{\beta}_{BC}(k|k) &= \arctan \frac{\hat{r}_{x,BC}(k|k)}{\hat{r}_{y,BC}(k|k)} \end{aligned}$$

Se  $|\hat{\beta}_{BC}(k) - \tilde{\beta}(k)| < \kappa\sigma_\eta$ :

$$\begin{aligned} \hat{\mathbf{c}}_{IV}(k) &= \left[ \cos \hat{\beta}_{BC}(k|k) \quad -\sin \hat{\beta}_{BC}(k|k) \quad 0 \quad 0 \right]^T \\ \mathbf{g}_{IV}(k) &= \mathbf{P}_e(k|k-1)\hat{\mathbf{c}}_{IV}(k) [\hat{\mathbf{c}}_{IV}^T(k)\mathbf{P}_e(k|k-1)\hat{\mathbf{c}}_{IV}(k) + P_\nu(k)]^{-1} \\ \hat{\mathbf{x}}_{IV}(k|k) &= \hat{\mathbf{x}}(k|k-1) - \mathbf{g}_{IV}(k)\hat{\mathbf{c}}_{IV}^T(k)\hat{\mathbf{x}}(k|k-1) \\ \mathbf{P}_{e,IV}(k|k) &= [\mathbf{I} - \mathbf{g}_{IV}(k)\hat{\mathbf{c}}_{IV}^T(k)] \mathbf{P}_e(k|k-1) \end{aligned}$$

Senão:  $\hat{\mathbf{x}}_{IV}(k|k) = \hat{\mathbf{x}}_{BC}(k|k)$ ;  $\mathbf{P}_{e,IV}(k|k) = \mathbf{P}_e(k|k)$

---

### 3.6 Aplicação Direta do Filtro de Kalman de Conjunto

O uso do EnKF consiste na aplicação direta do espaço de estados definido na Seção 3.1 nas equações da Subseção 2.1.3. Neste caso, são válidas as mesmas considerações feitas para o EKF, na Seção 3.2. Não foi encontrada referência discutindo a aplicação direta deste algoritmo, apenas aparecendo nos resultados comparativos do estudo de SUN e XIN [9]. Um resumo pode ser visto na Tabela 3.5.

### 3.7 Filtro de Kalman de Conjunto Aumentado

SUN e XIN [9] identificaram que, embora os modelos de filtro derivados do EKF aumentem a acurácia do resultado comparado ao EKF original, eles estão limitados pelo mínimo do erro quadrático médio linear (LMMSE – *Linear Minimum Mean*

Tabela 3.5: Filtro de Kalman de Conjunto aplicado a BOT

**EnKF aplicado a BOT** $\hat{\mathbf{x}}(0|0)$  estimadoPara  $k > 0$ :

$$\mathbf{w}(k) = \left[ \frac{\tau}{2} [\mathbf{v}_o^T(k) - \mathbf{v}_o^T(k-1)] \quad [\mathbf{v}_o^T(k) - \mathbf{v}_o^T(k-1)] \right]^T$$

$$\bar{\mathbf{x}}(k|k-1) = \mathbf{0}; \bar{\mathbf{y}}(k) = 0$$

$$\bar{\mathbf{P}}_e(k|k-1) = \mathbf{0}; \bar{P}_\beta(k) = 0$$

Para  $j$  de 1 a  $J$ :Gera  $\mathbf{n}_j(k)$  e  $\eta_j(k)$ 

$$\hat{\mathbf{x}}_j(k|k-1) = \mathbf{A}\hat{\mathbf{x}}(k-1|k-1) - \mathbf{w}(k) + \mathbf{n}_j(k)$$

$$\bar{\mathbf{x}}(k|k-1) = \bar{\mathbf{x}}(k|k-1) + \hat{\mathbf{x}}_j(k|k-1)$$

$$\hat{\beta}_j(k) = \tilde{\beta}(k) + \eta_j(k)$$

$$\bar{\beta}(k) = \bar{\beta}(k) + \hat{\beta}_j(k)$$

$$\bar{\mathbf{x}}(k|k-1) = \frac{1}{J}\bar{\mathbf{x}}(k|k-1)$$

$$\bar{\beta}(k) = \frac{1}{J}\bar{\beta}(k)$$

Para  $j$  de 1 a  $J$ :

$$\bar{\mathbf{P}}_e(k|k-1) = \bar{\mathbf{P}}_e(k|k-1) + [\hat{\mathbf{x}}_j(k|k-1) - \bar{\mathbf{x}}(k|k-1)][\hat{\mathbf{x}}_j(k|k-1) - \bar{\mathbf{x}}(k|k-1)]^T$$

$$\bar{P}_\beta(k) = \bar{P}_\beta(k) + [\hat{\beta}_j(k) - \bar{\beta}(k)]^2$$

$$\bar{\mathbf{P}}_e(k|k-1) = \frac{1}{J-1}\bar{\mathbf{P}}_e(k|k-1)$$

$$\mathbf{P}_e(k|k-1) = \mathbf{A}\mathbf{P}_e(k-1|k-1)\mathbf{A}^T + \bar{\mathbf{P}}_e(k|k-1)$$

$$\bar{P}_\beta(k) = \frac{1}{J-1}\bar{P}_\beta(k)$$

$$\mathbf{c}(k) = \left[ \frac{\bar{r}_y(k|k-1)}{\bar{r}_x^2(k|k-1) + \bar{r}_y^2(k|k-1)} \quad \frac{-\bar{r}_x(k|k-1)}{\bar{r}_x^2(k|k-1) + \bar{r}_y^2(k|k-1)} \quad 0 \quad 0 \right]^T$$

$$\mathbf{g}(k) = \mathbf{P}_e(k|k-1)\mathbf{c}(k) [\mathbf{c}^T(k)\mathbf{P}_e(k|k-1)\mathbf{c}(k) + \bar{P}_\beta(k)]^{-1}$$

$$\hat{\mathbf{x}}(k|k) = \bar{\mathbf{x}}(k|k-1) + \mathbf{g}(k) \left[ \bar{\beta}(k) - \arctan \frac{\bar{r}_x(k|k-1)}{\bar{r}_y(k|k-1)} \right]$$

$$\mathbf{P}_e(k|k) = [\mathbf{I} - \mathbf{g}(k)\mathbf{c}^T(k)] \mathbf{P}_e(k|k-1)$$

*Squared Error*) nos casos de sistemas de maior não-linearidade (ou também não-Gaussiano).

Segundo os autores, de forma geral, o LMMSE busca estimar  $\hat{\mathbf{x}}(k|k)$  com base na inovação trazida pela medição  $\mathbf{y}(k) - \hat{\mathbf{y}}(k) = \mathbf{y}(k) - \mathbf{C}^T(k)\hat{\mathbf{x}}(k|k-1)$ , conforme (2.8), onde o Ganho de Kalman é o coeficiente ótimo da inovação, obtido pela minimização do MSE de  $\hat{\mathbf{x}}$ . Mesmo os demais modelos não lineares ficam restritos à premissa de que o valor ótimo de  $\hat{\mathbf{x}}$  é uma função linear de  $\mathbf{y}$ . Para isso, é necessário generalizar o LMMSE (GLMMSE). A generalização busca uma solução do tipo:

$$\hat{\mathbf{x}}(k|k) = \arg \min_{A, B} \text{MSE} [A + B\mathbf{Y}_{aug}(k)] \quad (3.35)$$

onde  $\mathbf{Y}_{aug}(k) = [\mathbf{y}^T(k) \quad \Phi_a^T[\mathbf{y}(k)]]^T$ , que inclui a medição original  $\mathbf{y}(k)$  e a medição

aumentada  $\Phi_a[\mathbf{y}]$ , esta última sendo uma transformação não-linear de  $\mathbf{y}(k)$ .

Para a definição da transformação, SUN e XIN [9] recorreram ao conceito de Conversão Descorrelacionada (UC – *Uncorrelated Conversion*), utilizando uma base ortogonal polinomial. A opção da base é por uma expansão arbitrária de caos polinomial [35], dado que esta expansão independe da probabilidade da distribuição dos elementos do conjunto (e sim dos momentos), o que não é válido para uma expansão de caos polinomial tradicional como a de Hemite ou Laguerre. A demonstração da não-correlação entre  $\mathbf{y}$  e os polinômios, assim como entre eles mesmos, é feita em [9].

Aplicando ao problema de BOT, o AEnKF requer uma equação de transição de estado do tipo (3.19), onde  $\mathbf{u}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$  pode ser visto como uma pequena perturbação aleatória do modelo (o que no BC-PLKF poderia ser interpretado como pequena alteração na rota do alvo), necessária para gerar o conjunto de previsão.

Alternativamente ao proposto na Subseção 2.1.3, a previsão inicial deve partir de um conjunto de  $\hat{\mathbf{x}}_j(k-1|k-1)$  dado por:

$$\hat{\mathbf{x}}_j(k|k-1) = \mathbf{A}\hat{\mathbf{x}}_j(k-1|k-1) - \mathbf{w}(k) + \mathbf{u}_j(k) \quad (3.36)$$

e, com isso, é calculado  $\bar{\mathbf{x}}(k|k-1)$  e  $\bar{\mathbf{P}}_e(k|k-1)$  usando (2.19) e (2.20). Embora SUN e XIN [9] tenham proposto o uso destas equações, o emprego da solução proposta por TIPPETT *et al.* [10] para o cálculo de  $\mathbf{P}_e(k|k-1)$  (2.24) mostrou uma maior convergência dos resultados e será adotado daqui pra frente, como no EnKF.

Na fase de análise, diferente do EnKF, é gerado o conjunto utilizando as estimativas com base nas previsões de  $\hat{\mathbf{x}}_j(k|k-1)$ :

$$\hat{\beta}_j(k) = \arctan \frac{\hat{r}_{x,j}(k|k-1)}{\hat{r}_{y,j}(k|k-1)} + \eta_j(k) \quad (3.37)$$

Para a UC são necessários os momentos em relação à origem dos dados normalizados  $\hat{\beta}_{j,norm}(k)$ , dados por:

$$\hat{\beta}_{j,norm}(k) = \frac{\hat{\beta}_j(k) - \bar{\beta}_j(k)}{\mu_2^{\frac{1}{2}} \left[ \hat{\beta}_j(k) \right]} \quad (3.38)$$

onde  $\mu_i \left[ \hat{\beta}_j(k) \right]$  é o momento central de ordem  $i$  (no caso, ordem 2, que é a variância) e  $\bar{\beta}_j(k)$  é a média do conjunto de  $\hat{\beta}_j(k)$ , equivalente ao momento em torno da origem  $m_i \left[ \hat{\beta}_j(k) \right]$  de primeira ordem. Sendo assim, é necessário calcular  $\bar{\beta}_j(k)$  (3.39) e  $\mu_2 \left[ \hat{\beta}_j(k) \right]$  (3.40) para, então, calcular  $\hat{\beta}_{j,norm}(k)$  (3.38). A partir dos dados normalizados, devem ser calculados os momentos em relação à origem,  $m_i \left[ \hat{\beta}_{j,norm}(k) \right]$ ,

para  $i \geq 3$ , usando (3.39) com  $\hat{\beta}_{j, norm}(k)$  no lugar de  $\hat{\beta}_j(k)$ .<sup>2</sup>

$$m_i \left[ \hat{\beta}_j(k) \right] = \frac{1}{J} \sum_{j=1}^J \left[ \hat{\beta}_j(k) \right]^i \quad (3.39)$$

$$\mu_i \left[ \hat{\beta}_j(k) \right] = \frac{1}{J} \sum_{j=1}^J \left[ \hat{\beta}_j(k) - \bar{\beta}(k) \right]^i \quad (3.40)$$

Sendo assim, a UC polinomial tem a forma:

$$\Phi_a \left[ \hat{\beta}_j(k) \right] = \left[ P^{(2)} \left[ \hat{\beta}_j(k) \right] \quad P^{(3)} \left[ \hat{\beta}_j(k) \right] \quad \dots \quad P^{(L)} \left[ \hat{\beta}_j(k) \right] \right]^T, \quad (3.41)$$

onde  $P^{(L)}$  é um polinômio de ordem  $L$  da forma:

$$P^{(L)} \left[ \hat{\beta}_j(k) \right] = \sum_{l=0}^L p_l^{(L)} \left\{ \frac{\hat{\beta}_j(k) - \bar{\beta}_j(k)}{\mu_2^{\frac{1}{2}} \left[ \hat{\beta}_j(k) \right]} \right\}^l = \sum_{l=0}^L p_l^{(L)} \left[ \hat{\beta}_{j, norm}(k) \right]^l \quad (3.42)$$

e os coeficientes  $p_l^{(L)}$  do polinômio de ordem  $L$  são dados pela Tabela 3.6.

Tabela 3.6: Coeficientes da expansão de caos polinomial arbitrária

	$P^{(0)}$	$P^{(1)}$	$P^{(2)}$	$P^{(3)}$
$p_0^{(L)}$	1	0	-1	$m_3'^2 - m_3'^3 + m_3' m_4' - m_5'$
$p_1^{(L)}$	-	1	$-m_3'$	$-m_3' m_5' + m_3'^2 - m_4' + m_3' m_4'$
$p_2^{(L)}$	-	-	1	$-m_3' m_4' + m_5' - m_3'$
$p_3^{(L)}$	-	-	-	$1 - m_3' + m_3'^2$
Para simplificar a notação, $m_i' = m_i \left[ \hat{\beta}_{j, norm}(k) \right]$				

Fonte: Adaptado de OLADYSHKIN e NOWAK [35]

SUN e XIN [9] utilizaram duas versões: uma com expansão até polinômio do 2º grau e outra indo até 3º grau, o que já exige calcular momento de ordem 5.

Os elementos do conjunto das variáveis de medição aumentada  $\hat{\mathbf{y}}_{j, aug}(k)$  são dados por:

$$\hat{\mathbf{y}}_{j, aug}(k) = \left[ \hat{\beta}_j(k) \quad \Phi_a^T \left[ \hat{\beta}_j(k) \right] \right]^T = \left[ \hat{\beta}_j(k) \quad P^{(2)} \left[ \hat{\beta}_j(k) \right] \quad \dots \quad P^{(L)} \left[ \hat{\beta}_j(k) \right] \right]^T, \quad (3.43)$$

<sup>2</sup>Em [9] e [35] é apresentada uma alternativa, onde, ao invés de normalizar os dados, calcula-se  $\mu_i \left[ \hat{\beta}_j(k) \right]$  para  $i \geq 3$  e se obtém  $m_i \left[ \hat{\beta}_{j, norm}(k) \right]$  pela relação:  $m_i \left[ \hat{\beta}_{j, norm}(k) \right] = \mu_i \left[ \hat{\beta}_j(k) \right] \mu_2^{-\frac{i}{2}} \left[ \hat{\beta}_j(k) \right]$ . A opção neste trabalho pela outra forma se dá ao fato que a implementação em MATLAB se torna mais simples dessa maneira.

com valor médio

$$\bar{\mathbf{y}}_{aug}(k) = \left[ \bar{\beta}(k) \quad \bar{\phi}_{a,\hat{\beta}}^T(k) \right]^T \quad (3.44)$$

e

$$\bar{\phi}_{a,\hat{\beta}}(k) = \frac{1}{J} \sum_{j=1}^J \phi_a \left[ \hat{\beta}_j(k) \right] . \quad (3.45)$$

O cálculo do ganho de Kalman é feito de forma alternativa, utilizando as matrizes de covariância cruzada de  $\hat{\mathbf{x}}_j(k|k-1)$  e  $\hat{\mathbf{y}}_{j,aug}(k)$ ,  $\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}_{aug}}(k)$ , e de covariância de  $\hat{\mathbf{y}}_{j,aug}(k)$ ,  $\mathbf{P}_{\hat{\mathbf{y}}_{aug}}(k)$ :

$$\mathbf{G}(k) = \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}_{aug}}(k) \mathbf{P}_{\hat{\mathbf{y}}_{aug}}^{-1}(k) , \quad (3.46)$$

$$\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}_{aug}}(k) = \frac{1}{J-1} \sum_{j=1}^J [\hat{\mathbf{x}}_j(k|k-1) - \bar{\mathbf{x}}(k|k-1)] [\hat{\mathbf{y}}_{j,aug}(k) - \bar{\mathbf{y}}_{aug}(k)]^T \quad (3.47)$$

e

$$\mathbf{P}_{\hat{\mathbf{y}}_{aug}}(k) = \frac{1}{J-1} \sum_{j=1}^J [\hat{\mathbf{y}}_{j,aug}(k) - \bar{\mathbf{y}}_{aug}(k)] [\hat{\mathbf{y}}_{j,aug}(k) - \bar{\mathbf{y}}_{aug}(k)]^T . \quad (3.48)$$

O cálculo da predição com inovação depende de que a medição  $\tilde{\beta}(k)$  também seja transformada. Para isso, calcula-se  $\mathbf{y}_{aug}(k)$  aplicando (3.43) com  $\tilde{\beta}(k)$  no lugar de  $\hat{\beta}_j(k)$  e mantendo-se os momentos de  $\hat{\beta}_j(k)$  para cálculo do polinômio, conforme

$$\mathbf{y}_{aug}(k) = \left[ \tilde{\beta}(k) \quad P^{(2)} \left[ \tilde{\beta}(k) \right] \quad \dots \quad P^{(L)} \left[ \tilde{\beta}(k) \right] \right]^T \quad (3.49)$$

e

$$P^{(L)} \left[ \tilde{\beta}(k) \right] = \sum_{l=0}^L p_l^{(L)} \left\{ \frac{\tilde{\beta}(k) - \bar{\beta}(k)}{\mu_2^{\frac{1}{2}} \left[ \hat{\beta}_j(k) \right]} \right\}^l . \quad (3.50)$$

Com isso, é possível calcular a estimativa  $\hat{\mathbf{x}}(k|k)$  e o conjunto de análise de  $\hat{\mathbf{x}}_j(k|k)$  para a próxima iteração do algoritmo aplicando (3.51) e (3.52), respectivamente.

$$\hat{\mathbf{x}}(k|k) = \bar{\mathbf{x}}(k|k-1) + \mathbf{G}(k) [\mathbf{y}_{aug}(k) - \bar{\mathbf{y}}_{aug}(k)] \quad (3.51)$$

$$\hat{\mathbf{x}}_j(k|k) = \hat{\mathbf{x}}_j(k|k-1) + \mathbf{G}(k) [\mathbf{y}_{aug}(k) - \hat{\mathbf{y}}_{j,aug}(k)] \quad (3.52)$$

O valor de  $\mathbf{P}_e(k|K)$  pode ser obtido pela relação:

$$\mathbf{P}_e(k|k) = \bar{\mathbf{P}}_e(k|k-1) - \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}_{aug}}(k) \mathbf{P}_{\hat{\mathbf{y}}_{aug}}^{-1}(k) \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}_{aug}}^T(k) . \quad (3.53)$$

Quanto à inicialização, são usados em [9] valores estimados para  $\hat{\mathbf{x}}_j(0|0)$ , sem discussão sobre o assunto. Nas simulações são utilizados valores aleatórios com o valor esperado sendo o valor real da variável e aplicado um desvio-padrão arbitrado

(em torno de 30% do valor esperado) para o sorteio aleatório da inicialização. Um resumo do algoritmo AEnKF é apresentado na Tabela 3.7.

### 3.8 Filtro de Kalman Estendido com Coordenada Polar Modificada

Com base na Figura 3.1, a escolha imediata de variáveis em coordenada polar para o problema de BOT seria um vetor de variável  $[\beta(t) \ r(t) \ \dot{\beta}(t) \ v(t)]^T$ , onde  $\dot{\beta}(t) = \frac{\partial\beta(t)}{\partial t}$  e  $v(t) = \dot{r}(t) = \frac{\partial r(t)}{\partial t}$ . No entanto, é demonstrado em [7] que, em trechos de velocidade relativa constante, a matriz de covariância do erro pode ter problemas de mau condicionamento se  $\dot{\beta}(t)$  for desprezível.

Sendo assim, foi necessário modificar as componentes da variável de estado. HOELZER *et al.* [7] propuseram como candidatas as variáveis  $\rho(t) = \ln r(t)$  e sua derivada em relação ao tempo  $\dot{\rho}(t) = \frac{\dot{r}(t)}{r(t)} = \frac{v(t)}{r(t)}$ , porém substituindo a primeira por  $\ell(t) = \frac{1}{r(t)}$ , que seria um termo que surge nas equações do modelo quando há a aceleração do navio-observador em manobra. Desta forma, aplicando esta modificação, o modelo utilizando a variável de estado  $\mathbf{z}(t)$  descrita em (3.54) ficou referenciado como sendo de “coordenada polar modificada” (MP-EKF)[7]. A componente  $\ell(t)$  e  $\dot{\rho}(t)$  impõem restrição de  $r(t) \neq 0$ , o que já ocorre, na aplicação prática.

$$\mathbf{z}(t) = \left[ \beta(t) \ \dot{\beta}(t) \ \frac{\dot{r}(t)}{r(t)} \ \frac{1}{r(t)} \right]^T = \left[ \beta(t) \ \dot{\beta}(t) \ \dot{\rho}(t) \ \ell(t) \right]^T \quad (3.54)$$

Para definir o espaço de estados, as equações do movimento relativo definidas em (3.5) são transformadas em coordenadas polares e trabalhadas, inicialmente, em sua forma diferencial [7], sendo escritas da forma:

$$\frac{\partial \mathbf{z}(t)}{\partial t} = \dot{\mathbf{z}}(t) = \begin{bmatrix} \dot{\beta}(t) \\ \ddot{\beta}(t) \\ \ddot{\rho}(t) \\ \dot{\ell}(t) \end{bmatrix} = \begin{bmatrix} \dot{\beta}(t) \\ -2\dot{\rho}(t)\dot{\beta}(t) \\ \dot{\beta}^2(t) - \dot{\rho}^2(t) \\ -\dot{\rho}(t)\ell(t) \end{bmatrix} - \begin{bmatrix} 0 \\ \ell a_{o_{\tan}}(t) \\ \ell a_{o_{\text{rad}}}(t) \\ 0 \end{bmatrix}, \quad (3.55)$$

onde

$$a_{o_{\tan}}(t) = a_{o_x}(t) \cos \beta(t) - a_{o_y}(t) \sin \beta(t) \quad (3.56)$$

e

$$a_{o_{\text{rad}}}(t) = a_{o_x}(t) \sin \beta(t) + a_{o_y}(t) \cos \beta(t) \quad (3.57)$$

são, respectivamente, as acelerações tangencial e radial do observador (sendo man-

Tabela 3.7: Algoritmo geral do Filtro de Kalman de Conjunto Aumentado

---

**AEnKF**

---

$\hat{\mathbf{x}}_j(0|0)$  e  $\mathbf{P}_e(k-1|k-1)$  estimados

Para  $k > 0$ :

$$\begin{aligned} \mathbf{w}(k) &= \left[ \frac{T}{2} [\mathbf{v}_o^T(k) - \mathbf{v}_o^T(k-1)] \quad [\mathbf{v}_o^T(k) - \mathbf{v}_o^T(k-1)] \right]^T \\ \bar{\mathbf{x}}(k|k-1) &= \mathbf{0}; \bar{\beta}(k) = \mathbf{0}; \bar{\phi}_{\alpha, \hat{\beta}}(k) = \mathbf{0}; \mu_2 [\hat{\beta}_j(k)] = \mathbf{0}; m'_i = 0 \\ \bar{\mathbf{P}}_e(k|k-1) &= \mathbf{0}; \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}_{aug}}(k) = \mathbf{0}; \mathbf{P}_{\hat{\mathbf{y}}_{aug}}(k) = \mathbf{0} \end{aligned}$$

Para  $j$  de 1 a  $J$ :

Gera  $\mathbf{u}_j(k)$  e  $\eta_j(k)$

$$\hat{\mathbf{x}}_j(k|k-1) = \mathbf{A}\hat{\mathbf{x}}(k-1|k-1) - \mathbf{w}(k) + \mathbf{u}_j(k)$$

$$\bar{\mathbf{x}}(k|k-1) = \bar{\mathbf{x}}(k|k-1) + \hat{\mathbf{x}}_j(k|k-1)$$

$$\hat{\beta}_j(k) = \arctan \frac{\hat{r}_{x,j}(k|k-1)}{\hat{r}_{y,j}(k|k-1)} + \eta_j(k)$$

$$\bar{\beta}(k) = \bar{\beta}(k) + \hat{\beta}_j(k)$$

$$\bar{\mathbf{x}}(k|k-1) = \frac{1}{J}\bar{\mathbf{x}}(k|k-1)$$

$$\bar{\beta}(k) = \frac{1}{J}\bar{\beta}(k)$$

Para  $j$  de 1 a  $J$ :  $\mu_2 [\hat{\beta}_j(k)] = \mu_2 [\hat{\beta}_j(k)] + [\hat{\beta}_j(k) - \bar{\beta}_j(k)]^2$

$$\mu_2 [\hat{\beta}_j(k)] = \frac{1}{J}\mu_2 [\hat{\beta}_j(k)]$$

Para  $j$  de 1 a  $J$ :

$$\hat{\beta}_{j, norm}(k) = [\hat{\beta}_j(k) - \bar{\beta}_j(k)] \mu_2^{-\frac{1}{2}} [\hat{\beta}_j(k)]$$

Para  $i$  de 3 a maior ordem de momento necessária:  $m'_i = m'_i + [\hat{\beta}_{j, norm}(k)]^i$

Para  $i$  de 3 a maior ordem de momento necessária:  $m'_i = \frac{1}{J}m'_i$

Calcular  $p_i^{(L)}$ , conforme Tabela 3.6

Para  $j$  de 1 a  $J$ :

Calcular  $P^{(L)} [\hat{\beta}_j(k)]$  e  $\phi_\alpha [\hat{\beta}_j(k)]$ , conforme (3.42) e (3.41)

$$\hat{\mathbf{y}}_{j, aug}(k) = \left[ \hat{\beta}_j(k) \quad \phi_\alpha^T(\hat{\beta}_j(k)) \right]^T$$

$$\bar{\phi}_{\alpha, \hat{\beta}}(k) = \bar{\phi}_{\alpha, \hat{\beta}}(k) + \phi_\alpha [\hat{\beta}_j(k)]$$

$$\bar{\phi}_{\alpha, \hat{\beta}}(k) = \frac{1}{J}\bar{\phi}_{\alpha, \hat{\beta}}(k)$$

$$\bar{\mathbf{y}}_{aug}(k) = \left[ \bar{\beta}(k) \quad \bar{\phi}_{\alpha, \hat{\beta}}^T(k) \right]^T$$

Para  $j$  de 1 a  $J$ :

$$\bar{\mathbf{P}}_e(k|k-1) = \bar{\mathbf{P}}_e(k|k-1) + [\hat{\mathbf{x}}_j(k|k-1) - \bar{\mathbf{x}}(k|k-1)] [\hat{\mathbf{x}}_j(k|k-1) - \bar{\mathbf{x}}(k|k-1)]^T$$

$$\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}_{aug}}(k) = \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}_{aug}}(k) + [\hat{\mathbf{x}}_j(k|k-1) - \bar{\mathbf{x}}(k|k-1)] [\hat{\mathbf{y}}_{j, aug}(k) - \bar{\mathbf{y}}_{aug}(k)]^T$$

$$\mathbf{P}_{\hat{\mathbf{y}}_{aug}}(k) = \mathbf{P}_{\hat{\mathbf{y}}_{aug}}(k) + [\hat{\mathbf{y}}_{j, aug}(k) - \bar{\mathbf{y}}_{aug}(k)] [\hat{\mathbf{y}}_{j, aug}(k) - \bar{\mathbf{y}}_{aug}(k)]^T$$

$$\bar{\mathbf{P}}_e(k|k-1) = \frac{1}{J}\bar{\mathbf{P}}_e(k|k-1) + \mathbf{A}\mathbf{P}_e(k-1|k-1)\mathbf{A}^T$$

$$\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}_{aug}}(k) = \frac{1}{J-1}\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}_{aug}}(k)$$

$$\mathbf{P}_{\hat{\mathbf{y}}_{aug}}(k) = \frac{1}{J-1}\mathbf{P}_{\hat{\mathbf{y}}_{aug}}(k)$$

$$\mathbf{G}(k) = \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}_{aug}}(k)\mathbf{P}_{\hat{\mathbf{y}}_{aug}}^{-1}(k)$$

Calcular  $P^{(L)} [\hat{\beta}(k)]$  e  $\mathbf{y}_{aug}(k)$ , conforme (3.50) e (3.49)

$$\hat{\mathbf{x}}(k|k) = \bar{\mathbf{x}}(k|k-1) + \mathbf{G}(k) [\mathbf{y}_{aug}(k) - \bar{\mathbf{y}}_{aug}(k)]$$

Para  $j$  de 1 a  $J$ :  $\hat{\mathbf{x}}_j(k|k) = \hat{\mathbf{x}}_j(k|k-1) + \mathbf{G}(k) [\mathbf{y}_{aug}(k) - \hat{\mathbf{y}}_{j, aug}(k)]$

$$\mathbf{P}_e(k|k) = \bar{\mathbf{P}}_e(k|k-1) - \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}_{aug}}(k)\mathbf{P}_{\hat{\mathbf{y}}_{aug}}^{-1}(k)\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}_{aug}}^T(k)$$


---

tida a premissa de MRU para o alvo).

Em coordenadas polares, a equação de medida torna-se simples e é dada por:

$$\beta(t) = \mathbf{c}^T \mathbf{z}(t) + \eta(t) , \quad (3.58)$$

com

$$\mathbf{c} = [1 \ 0 \ 0 \ 0]^T . \quad (3.59)$$

O uso de coordenadas polares faz com que a não linearidade do problema apareça na equação de estado, e não mais na equação de medição. Com base nas equações descritas na Subseção 2.1.2, o erro causado na matriz de covariância devido à linearização se torna independente do erro do cálculo da distância durante a manobra do navio-observador [7].

Para linearizar a equação de transição de estados e discretizar o tempo, HOELZER *et al.* [7] propõem separar a manobra da transição temporal em MRU. Sendo assim, considera-se a aceleração no momento da manobra como uma força impulsiva que afeta apenas as variáveis de velocidade do navio-observador, ou seja,  $\dot{\beta}(t)$  e  $\dot{\rho}(t)$ . Discretizando no tempo, para atualizar  $\mathbf{z}(k)$  com base em  $\mathbf{z}(k-1)$ , considera-se toda a variação de velocidade entre os instantes  $k-1$  e  $k$  como um impulso atuando do instante  $(k-1)^-$  ao  $(k-1)^+$ , que modificará as componentes  $\dot{\rho}(t)$  e  $\dot{\beta}(t)$  de  $\mathbf{z}(k-1)$ , sendo esta variável atualizada  $\mathbf{z}(k-1)^+$  aplicada para fazer a transição temporal em velocidade constante.

A variação de velocidade provocada pela manobra pode ser entendida pela variação de  $\dot{\rho}(t)$  e  $\dot{\beta}(t)$ . Sendo  $v_{\text{tan}}$  a velocidade tangencial relativa e  $v_{\text{rad}}$  a velocidade radial relativa, e  $v_{o_{\text{tan}}}$  e  $v_{o_{\text{rad}}}$  as velocidades tangencial e radial do observador, temos que:

$$\begin{aligned} v_{\text{tan}}(t) &= \dot{\beta}(t)r(t) = \frac{\dot{\beta}(t)}{\ell(t)} \Leftrightarrow \dot{\beta}(t) = \ell(t)v_{\text{tan}}(t) \\ \dot{\beta}(t) - \dot{\beta}(t_0) &= \ell(t_0)[v_{\text{tan}}(t) - v_{\text{tan}}(t_0)] = -\ell(t_0)[v_{o_{\text{tan}}}(t) - v_{o_{\text{tan}}}(t_0)] \end{aligned} \quad (3.60)$$

$$\begin{aligned} v_{\text{rad}}(t) &= \dot{\rho}(t) = \frac{\dot{\rho}(t)}{\ell(t)} \Leftrightarrow \dot{\rho}(t) = \ell(t)v_{\text{rad}}(t) \\ \dot{\rho}(t) - \dot{\rho}(t_0) &= \ell(t_0)[v_{\text{rad}}(t) - v_{\text{rad}}(t_0)] = -\ell(t_0)[v_{o_{\text{rad}}}(t) - v_{o_{\text{rad}}}(t_0)] \end{aligned} \quad (3.61)$$

Logo, o efeito da manobra depende apenas da variação de velocidade do observador, que é conhecida, e sua transformação para os referenciais tangencial e radial é dada por:

$$v_{o_{\text{tan}}}(t) = v_{o_x}(t) \cos \beta(t) - v_{o_y}(t) \sin \beta(t) \quad (3.62)$$

e

$$v_{o_{\text{rad}}}(t) = v_{o_x}(t) \sin \beta(t) + v_{o_y}(t) \cos \beta(t) . \quad (3.63)$$

A transição, já discretizada no tempo com  $t = k\tau$ , pode ser escrita como:

$$\hat{\mathbf{z}}(k-1|k-1)^+ = \mathbf{A}_\delta(k, k-1)\hat{\mathbf{z}}(k-1|k-1)^-, \quad (3.64)$$

onde

$$\mathbf{A}_\delta(k, k-1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & v_{o_{\tan}}(k-1) - v_{o_{\tan}}(k) \\ 0 & 0 & 1 & v_{o_{\text{rad}}}(k-1) - v_{o_{\text{rad}}}(k) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.65)$$

A equação de transição de estados sem manobra, após manipulação algébrica de (3.55), com  $a_{o_{\tan}} = a_{o_{\text{rad}}} = 0$  [7], e discretizada no tempo, pode ser escrita como a função:

$$\mathbf{z}(k) = \mathbf{f}[\mathbf{z}(k-1)] = \begin{bmatrix} \beta(k-1) + \arctan \frac{\dot{\beta}(k-1)\tau}{1 + \dot{\rho}(k-1)\tau} \\ \frac{\dot{\beta}(k-1)}{\mathcal{D}(k-1)} \\ \frac{\dot{\rho}(k-1) + [\dot{\rho}^2(k-1) + \dot{\beta}^2(k-1)]\tau}{\mathcal{D}(k-1)} \\ \frac{\ell(k-1)}{\mathcal{D}^{\frac{1}{2}}(k-1)} \end{bmatrix}, \quad (3.66)$$

onde:

$$\mathcal{D}(k-1) = [1 + \dot{\rho}(k-1)\tau]^2 + [\dot{\beta}(k-1)\tau]^2. \quad (3.67)$$

Desta forma, a transição é feita em duas etapas: primeiro, aplica-se (3.64) para se obter  $\hat{\mathbf{z}}(k-1|k-1)^+$ ; depois, calcula-se  $\hat{\mathbf{z}}(k|k-1) = \mathbf{f}[\hat{\mathbf{z}}(k-1|k-1)^+]$  aplicando-se (3.66).

O mesmo raciocínio deve ser aplicado para a matriz de correlação do erro  $\mathbf{P}_e(k|k-1)$ . Neste caso, como a função de transição de estado não é linear, aplica-se o conceito do EKF de aproximação pela primeira derivada (2.15) para determinar a matriz de transição de estados sem manobra  $\mathbf{A}_{\text{MRU}}(k-1)$ :

$$\mathbf{A}_{\text{MRU}}(k-1) = \begin{bmatrix} 1 & \frac{[1 + \dot{\rho}(k-1)\tau]\tau}{\mathcal{D}(k-1)} & \frac{-[\dot{\beta}(k-1)\tau]\tau}{\mathcal{D}(k-1)} & 0 \\ 0 & \frac{[1 + \dot{\rho}(k-1)\tau]^2 - [\dot{\beta}(k-1)\tau]^2}{\mathcal{D}^2(k-1)} & \frac{-2[\dot{\beta}(k-1)\tau][1 + \dot{\rho}(k-1)\tau]}{\mathcal{D}^2(k-1)} & 0 \\ 0 & \frac{2[\dot{\beta}(k-1)\tau][1 + \dot{\rho}(k-1)\tau]}{\mathcal{D}^2(k-1)} & \frac{[1 + \dot{\rho}(k-1)\tau]^2 - [\dot{\beta}(k-1)\tau]^2}{\mathcal{D}^2(k-1)} & 0 \\ 0 & \frac{-[\dot{\beta}(k-1)\tau][\ell(k-1)\tau]}{\mathcal{D}^{\frac{3}{2}}(k-1)} & \frac{-[1 + \dot{\rho}(k-1)\tau][\ell(k-1)\tau]}{\mathcal{D}^{\frac{3}{2}}(k-1)} & \frac{1}{\mathcal{D}^{\frac{1}{2}}(k-1)} \end{bmatrix}. \quad (3.68)$$

Com essa linearização, o cálculo de  $\mathbf{P}_e(k|k-1)$  (2.4), aqui indicado por

$\mathbf{P}_{e,MP}(k|k-1)$ , pode ser feito pela combinação das matrizes de transição  $\mathbf{A}_\delta(k, k-1)$  e  $\mathbf{A}_{MRU}(k-1)$ :

$$\mathbf{P}_{e,MP}(k|k-1) = \mathbf{A}_{MRU}(k-1)\mathbf{A}_\delta(k, k-1)\mathbf{P}_{e,MP}(k-1|k-1)\mathbf{A}_\delta^T(k, k-1)\mathbf{A}_{MRU}^T(k-1) \quad (3.69)$$

e, a partir deste ponto, o algoritmo pode seguir com o cálculo do ganho e demais passos do EKF, já apresentados na Tabela 2.2.

AIDALA e HAMMEL [21] (e outros autores que vieram depois, como PEACH [23] e JAWAHAR e KOTESWARA RAO [26]) apresentaram um algoritmo unificado, onde não há a quebra da transição de estado em impulso (aceleração) e deslocamento. No entanto, esta abordagem trouxe junto uma maior complexidade algébrica das equações. Esta formulação de transição única também é aplicada ao LP-EKF e será abordada na Seção 3.9.

A conversão da variável cartesiana  $\mathbf{x}(k)$  (3.4) para a polar modificada  $\mathbf{z}(k)$  (3.54) é dada pela transformação  $\boldsymbol{\psi}_{MP}[\mathbf{x}(k)]$  (3.70) e sua inversa pela transformação  $\boldsymbol{\psi}_{MP}^{-1}[\mathbf{z}(k)]$  (3.71).

$$\mathbf{z}(k) = \boldsymbol{\psi}_{MP}[\mathbf{x}(k)] = \begin{bmatrix} \arctan \frac{r_x(k)}{r_y(k)} \\ \frac{v_x(k)r_y(k) - v_y(k)r_x(k)}{r_x^2(k) + r_y^2(k)} \\ \frac{v_x(k)r_x(k) + v_y(k)r_y(k)}{r_x^2(k) + r_y^2(k)} \\ \frac{1}{\sqrt{r_x^2(k) + r_y^2(k)}} \end{bmatrix} \quad (3.70)$$

$$\mathbf{x}(k) = \boldsymbol{\psi}_{MP}^{-1}[\mathbf{z}(k)] = \frac{1}{\ell(k)} \begin{bmatrix} \sin \beta(k) \\ \cos \beta(k) \\ \dot{\rho}(k) \sin \beta(k) + \dot{\beta}(k) \cos \beta(k) \\ \dot{\rho}(k) \cos \beta(k) - \dot{\beta}(k) \sin \beta(k) \end{bmatrix} \quad (3.71)$$

Com relação à inicialização, HOELZER *et al.* [7] propõem uma inicialização mais realística. Para a variável de estado  $\hat{\mathbf{z}}(0|0)$ , é proposto  $\hat{\beta}(0|0) = \tilde{\beta}(0)$ ,  $\hat{\dot{\beta}}(0|0) = \hat{\dot{\rho}}(0|0) = 0$  (uma vez que não se sabe o rumo do alvo) e, para  $\hat{\ell}(0|0)$ , que se use uma distância média de detecção de 15 nmi (milhas náuticas)<sup>3</sup>, fazendo então  $\hat{\ell}(0|0) = 1/15 \text{ nmi}^{-1}$ . Para a inicialização da matriz de covariância  $\mathbf{P}_{e,MP}(0|0)$ , é proposto: para a incerteza de  $\hat{\beta}(0|0)$ , usar  $\sigma_\eta(0)$ ; para as incertezas de  $\hat{\dot{\beta}}(0|0)$  e  $\hat{\dot{\rho}}(0|0)$ , utilizar uma razão entre máxima velocidade sobre mínima distância; e, para incerteza de  $\hat{\ell}(0|0)$ , usar o inverso da mínima distância. AIDALA e HAMMEL [21] e PEACH

<sup>3</sup>1 milha náutica corresponde a 1.852 metros exatos [3]; não há uma simbologia padronizada para a unidade, sendo as mais usadas nmi, nm ou mn.

[23] propõem uma matriz simplificada utilizando as estimativas de distância ao alvo  $\hat{r}(0)$  e os desvios-padrão da distância  $\sigma_{\hat{r}}$  e da velocidade  $\sigma_{\hat{v}_t}$  do alvo:

$$\mathbf{P}_{e,MP}(0|0) = \text{diag} \left[ \sigma_{\eta}^2 \quad \frac{\sigma_{\hat{v}_t}^2}{\hat{r}^2(0)} \quad \frac{\sigma_{\hat{v}_t}^2}{\hat{r}^2(0)} \quad \frac{\sigma_{\hat{r}}^2}{\hat{r}^4(0)} \right]. \quad (3.72)$$

Já JAWAHAR e KOTESWARA RAO [26] propõem uma inicialização tradicional baseada em valores esperados e seus desvios, como indicado nas Subseção 2.1.1 e Seção 3.2, propondo também uma forma de conversão da estimativa de  $\mathbf{P}_e(0|0)$  (3.14) para coordenada polar modificada:

$$\mathbf{P}_{e,MP}(0|0) = \mathbb{E} \left\{ [\hat{\mathbf{z}}(0|0) - \mathbf{z}(0)] [\hat{\mathbf{z}}(0|0) - \mathbf{z}(0)]^T \right\} \approx \mathbf{\Psi}_{MP} \mathbf{P}_e(0|0) \mathbf{\Psi}_{MP}^T, \quad (3.73)$$

onde

$$\begin{aligned} \mathbf{\Psi}_{MP} &= \left. \frac{\partial \psi_{MP}[\mathbf{x}(k)]}{\partial \mathbf{x}(k)} \right|_{\mathbf{x}(k)=\hat{\mathbf{x}}(0|0)} \\ &= \begin{bmatrix} \frac{\hat{r}_y}{\hat{r}_x^2 + \hat{r}_y^2} & \frac{-\hat{r}_x}{\hat{r}_x^2 + \hat{r}_y^2} & 0 & 0 \\ \frac{\hat{v}_y(\hat{r}_x^2 - \hat{r}_y^2) - 2v_x r_x r_y}{(\hat{r}_x^2 + \hat{r}_y^2)^2} & \frac{\hat{v}_x(\hat{r}_x^2 - \hat{r}_y^2) + 2v_y r_x r_y}{(\hat{r}_x^2 + \hat{r}_y^2)^2} & \frac{\hat{r}_y}{\hat{r}_x^2 + \hat{r}_y^2} & \frac{-\hat{r}_x}{\hat{r}_x^2 + \hat{r}_y^2} \\ -\frac{\hat{v}_x(\hat{r}_x^2 - \hat{r}_y^2) + 2v_y r_x r_y}{(\hat{r}_x^2 + \hat{r}_y^2)^2} & \frac{\hat{v}_y(\hat{r}_x^2 - \hat{r}_y^2) - 2v_x r_x r_y}{(\hat{r}_x^2 + \hat{r}_y^2)^2} & \frac{\hat{r}_x}{\hat{r}_x^2 + \hat{r}_y^2} & \frac{\hat{r}_y}{\hat{r}_x^2 + \hat{r}_y^2} \\ \frac{-\hat{r}_x}{(\hat{r}_x^2 + \hat{r}_y^2)^{\frac{3}{2}}} & \frac{-\hat{r}_y}{(\hat{r}_x^2 + \hat{r}_y^2)^{\frac{3}{2}}} & 0 & 0 \end{bmatrix}. \end{aligned} \quad (3.74)$$

Em (3.74) foram omitidos intencionalmente os indicadores temporais “(0|0)” das variáveis no interior da matriz para não dificultar a compreensão. Vale ressaltar que a matriz possui elementos iguais ou simétricos, simplificando a implementação do cálculo.

Um resumo do MP-EKF pode visto na Tabela 3.8.

### 3.9 Filtro de Kalman Estendido com Coordenada Log-Polar

Conforme visto na Seção 3.8, o MP-EKF, definido em [7], optou por utilizar a variável  $\ell(t) = \frac{1}{r(t)}$  no lugar de  $\rho(t) = \ln r(t)$ . Posteriormente, trabalhos como de BREHARD e CADRE [24] e LA SCALA e MORELANDE [25] resgataram a variável  $\rho(t)$  e identificaram vantagens como maior robustez em alguns cenários, mas, principalmente, em calcular o CRLB. Com isso, o uso da variável de estado em

---

Tabela 3.8: Filtro de Kalman Estendido com Coordenada Polar Modificada

---

**MP-EKF**

---

$\hat{\mathbf{z}}(0|0)$  e  $\mathbf{P}_{e,MP}(0|0)$  estimados

Para  $k > 0$ :

$$\begin{aligned}
 \mathbf{A}_\delta(k, k-1) &= [\text{conforme (3.65)}] \\
 \hat{\mathbf{z}}(k-1|k-1)^+ &= \mathbf{A}_\delta(k, k-1)\hat{\mathbf{z}}(k-1|k-1)^- \\
 \mathcal{D}(k-1|k-1) &= [1 + \dot{\rho}(k-1|k-1)^+ \tau]^2 + [\dot{\beta}(k-1|k-1)^+ \tau]^2 \\
 \hat{\mathbf{z}}(k|k-1) &= \mathbf{f}[\hat{\mathbf{z}}(k-1|k-1)^+], \text{ conforme (3.66)} \\
 \mathbf{A}_{\text{MRU}}(k-1) &= [\text{conforme (3.68)}] \\
 \mathbf{P}_{e,MP}(k|k-1) &= \mathbf{A}_{\text{MRU}}(k-1)\mathbf{A}_\delta(k, k-1)\mathbf{P}_{e,MP}(k-1|k-1)\mathbf{A}_\delta^T(k, k-1)\mathbf{A}_{\text{MRU}}^T(k-1) \\
 \mathbf{g}(k) &= \mathbf{P}_{e,MP}(k|k-1)\mathbf{c}(k) [\mathbf{c}^T(k)\mathbf{P}_{e,MP}(k|k-1)\mathbf{c}(k) + \sigma_\eta^2(k)]^{-1} \\
 \hat{\mathbf{z}}(k|k) &= \hat{\mathbf{z}}(k|k-1) + \mathbf{g}(k) [\tilde{\beta}(k) - \mathbf{c}^T\hat{\mathbf{z}}(k|k-1)] \\
 \mathbf{P}_{e,MP}(k|k) &= [\mathbf{I} - \mathbf{g}(k)\mathbf{c}^T(k)] \mathbf{P}_{e,MP}(k|k-1)
 \end{aligned}$$


---

(3.75) foi referenciado como “coordenada log-polar”, que deu o nome ao LP-EKF. Assim como para o MP-EKF, o LP-EKF impõe restrição de  $r(t) \neq 0$ .

$$\mathbf{z}(t) = \left[ \beta(t) \quad \ln r(t) \quad \dot{\beta}(t) \quad \frac{\dot{r}(t)}{r(t)} \right]^T = \left[ \beta(t) \quad \rho(t) \quad \dot{\beta}(t) \quad \dot{\rho}(t) \right]^T \quad (3.75)$$

A conversão da variável cartesiana  $\mathbf{x}(k)$  (3.4) para a log-polar  $\mathbf{z}(k)$  (3.75) é dada pela relação trigonométrica das variáveis, sendo representada pela transformação  $\boldsymbol{\psi}_{LP}[\mathbf{x}(k)]$  (3.76) e sua inversa pela transformação  $\boldsymbol{\psi}_{LP}^{-1}[\mathbf{z}(k)]$  (3.77).

$$\mathbf{z}(k) = \boldsymbol{\psi}_{LP}[\mathbf{x}(k)] = \begin{bmatrix} \arctan \frac{r_x(k)}{r_y(k)} \\ \frac{1}{2} \ln [r_x^2(k) + r_y^2(k)] \\ \frac{v_x(k)r_y(k) - v_y(k)r_x(k)}{r_x^2(k) + r_y^2(k)} \\ \frac{v_x(k)r_x(k) + v_y(k)r_y(k)}{r_x^2(k) + r_y^2(k)} \end{bmatrix} \quad (3.76)$$

$$\mathbf{x}(k) = \boldsymbol{\psi}_{LP}^{-1}[\mathbf{z}(k)] = e^{\rho(k)} \begin{bmatrix} \sin \beta(k) \\ \cos \beta(k) \\ \dot{\rho}(k) \sin \beta(k) + \dot{\beta}(k) \cos \beta(k) \\ \dot{\rho}(k) \cos \beta(k) - \dot{\beta}(k) \sin \beta(k) \end{bmatrix} \quad (3.77)$$

Para definir o espaço de estados, as equações do movimento relativo definidas em (3.5) são mapeadas em coordenada log-polar, similar ao encontrado em [21] para

polar modificada. Sendo assim, a equação de transição de estados é dada por [25]:

$$\mathbf{z}(k) = \mathbf{f}[\mathbf{z}(k-1)] = \begin{bmatrix} \beta(k-1) + \arctan \frac{\lambda_1(k)}{\lambda_2(k)} \\ \rho(k-1) + \frac{1}{2} \ln [\lambda_1^2(k) + \lambda_2^2(k)] \\ \frac{\lambda_3(k)\lambda_2(k) - \lambda_4(k)\lambda_1(k)}{\lambda_1^2(k) + \lambda_2^2(k)} \\ \frac{\lambda_3(k)\lambda_1(k) + \lambda_4(k)\lambda_2(k)}{\lambda_1^2(k) + \lambda_2^2(k)} \end{bmatrix}, \quad (3.78)$$

onde:

$$\begin{aligned} \lambda_1(k) &= \tau \dot{\beta}(k-1) - \frac{\tau}{2} e^{-\rho(k-1)} [\Delta v_{o_x}(k) \cos \beta(k-1) - \Delta v_{o_y}(k) \sin \beta(k-1)] \\ \lambda_2(k) &= 1 + \tau \dot{\rho}(k-1) - \frac{\tau}{2} e^{-\rho(k-1)} [\Delta v_{o_x}(k) \sin \beta(k-1) + \Delta v_{o_y}(k) \cos \beta(k-1)] \\ \lambda_3(k) &= \dot{\beta}(k-1) - e^{-\rho(k-1)} [\Delta v_{o_x}(k) \cos \beta(k-1) - \Delta v_{o_y}(k) \sin \beta(k-1)] \\ \lambda_4(k) &= \dot{\rho}(k-1) - e^{-\rho(k-1)} [\Delta v_{o_x}(k) \sin \beta(k-1) + \Delta v_{o_y}(k) \cos \beta(k-1)] \\ \Delta v_{o_x}(k) &= v_{o_x}(k) - v_{o_x}(k-1) \\ \Delta v_{o_y}(k) &= v_{o_y}(k) - v_{o_y}(k-1) . \end{aligned} \quad (3.79)$$

Em coordenada log-polar, a equação de medição também se torna simples e é dada por:

$$\beta(k) = \mathbf{c}^T \mathbf{z}(k) + \eta(k) \quad (3.80)$$

e

$$\mathbf{c} = [1 \ 0 \ 0 \ 0]^T . \quad (3.81)$$

Assim como o MP-EKF, o uso de coordenada log-polar faz com que a não linearidade do problema seja na equação de estado, e não mais na equação de medição. Com base nas equações descritas na Subseção 2.1.2, o erro causado na matriz de covariância devido à linearização se torna independente do erro do cálculo da distância durante a manobra do navio-observador [7].

Com isso, para aplicar o EKF com esta coordenada, é necessário linearizar (3.78) aplicando-se (2.15). O cálculo de  $\mathbf{A}(k-1)$  encontra-se no Apêndice A. Assim, é possível continuar com as etapas do algoritmo descrito na Subseção 2.1.2.

Com relação à inicialização, já foi dito que LA SCALA e MORELANDE [25] propõe o uso de parâmetros estimados, como descrito na Seção 3.2, estimando as variáveis em coordenada cartesiana e convertendo, posteriormente, para log-polar. A conversão de  $\mathbf{P}_e(0|0)$  para log-polar é feita por:

$$\mathbf{P}_{e,LP}(0|0) = \mathbb{E} \left\{ [\hat{\mathbf{z}}(0|0) - \mathbf{z}(0)] [\hat{\mathbf{z}}(0|0) - \mathbf{z}(0)]^T \right\} \approx \mathbf{\Psi}_{LP} \mathbf{P}_e(0|0) \mathbf{\Psi}_{LP}^T \quad (3.82)$$

onde

$$\Psi_{LP} = \left. \frac{\partial \psi_{LP}[\mathbf{x}(k)]}{\partial \mathbf{x}(k)} \right|_{\mathbf{x}(k)=\hat{\mathbf{x}}(0|0)} = \begin{bmatrix} \frac{\hat{r}_y}{\hat{r}_x^2 + \hat{r}_y^2} & \frac{-\hat{r}_x}{\hat{r}_x^2 + \hat{r}_y^2} & 0 & 0 \\ \frac{\hat{r}_x}{\hat{r}_x^2 + \hat{r}_y^2} & \frac{\hat{r}_y}{\hat{r}_x^2 + \hat{r}_y^2} & 0 & 0 \\ \frac{\hat{v}_y(\hat{r}_x^2 - \hat{r}_y^2) - 2v_x r_x r_y}{(\hat{r}_x^2 + \hat{r}_y^2)^2} & \frac{\hat{v}_x(\hat{r}_x^2 - \hat{r}_y^2) + 2v_y r_x r_y}{(\hat{r}_x^2 + \hat{r}_y^2)^2} & \frac{\hat{r}_y}{\hat{r}_x^2 + \hat{r}_y^2} & \frac{-\hat{r}_x}{\hat{r}_x^2 + \hat{r}_y^2} \\ \frac{\hat{v}_x(\hat{r}_x^2 - \hat{r}_y^2) + 2v_y r_x r_y}{(\hat{r}_x^2 + \hat{r}_y^2)^2} & \frac{\hat{v}_y(\hat{r}_x^2 - \hat{r}_y^2) - 2v_x r_x r_y}{(\hat{r}_x^2 + \hat{r}_y^2)^2} & \frac{\hat{r}_x}{\hat{r}_x^2 + \hat{r}_y^2} & \frac{\hat{r}_y}{\hat{r}_x^2 + \hat{r}_y^2} \end{bmatrix}. \quad (3.83)$$

Em (3.83) foram omitidos intencionalmente os indicadores temporais “(0|0)” das variáveis na matriz para não dificultar a compreensão. Vale ressaltar que a matriz possui elementos iguais ou simétricos, simplificando a implementação do cálculo.

Ainda referente à inicialização, é possível utilizar as mesmas propostas de HOELZER *et al.* [7], AIDALA e HAMMEL [21] e PEACH [23] descritas na Seção 3.8, fazendo a devida substituição de  $\ell(k)$  para  $\rho(k)$ . Em especial, (3.72) se torna:

$$\mathbf{P}_{e,LP}(0|0) = \text{diag} \left[ \sigma_\eta^2 \quad \frac{\sigma_{\hat{r}}^2}{\hat{r}^2(0)} \quad \frac{\sigma_{\hat{v}_t}^2}{\hat{r}^2(0)} \quad \frac{\sigma_{\hat{v}_t}^2}{\hat{r}^2(0)} \right]. \quad (3.84)$$

Um resumo do LP-EKF pode visto na Tabela 3.9.

Tabela 3.9: Filtro de Kalman Estendido com Coordenada Log-Polar

---

**LP-EKF**

---

$\hat{\mathbf{z}}(0|0)$  e  $\mathbf{P}_{e,LP}(0|0)$  estimados

Para  $k > 0$ :

$$\hat{\mathbf{z}}(k|k-1) = \mathbf{f}[\hat{\mathbf{z}}(k-1|k-1)], \text{ conforme (3.78)}$$

$$\mathbf{A}(k-1) = [\text{conforme Apêndice A}]$$

$$\mathbf{P}_{e,LP}(k|k-1) = \mathbf{A}(k-1)\mathbf{P}_{e,LP}(k-1|k-1)\mathbf{A}^T(k-1)$$

$$\mathbf{g}(k) = \mathbf{P}_{e,LP}(k|k-1)\mathbf{c}(k) \left[ \mathbf{c}^T(k)\mathbf{P}_{e,LP}(k|k-1)\mathbf{c}(k) + \sigma_\eta^2(k) \right]^{-1}$$

$$\hat{\mathbf{z}}(k|k) = \hat{\mathbf{z}}(k|k-1) + \mathbf{g}(k) \left[ \tilde{\beta}(k) - \mathbf{c}^T \hat{\mathbf{z}}(k|k-1) \right]$$

$$\mathbf{P}_{e,LP}(k|k) = [\mathbf{I} - \mathbf{g}(k)\mathbf{c}^T(k)] \mathbf{P}_{e,LP}(k|k-1)$$


---

### 3.10 Filtro de Kalman Estendido Aplicado a BOT com Parametrização de Distância e Velocidade

Diferente da proposta do EnKF, onde se trabalha com um conjunto de variáveis de estado que se diferem por uma perturbação aleatória, PEACH [23] propõe em sua tese de doutorado a utilização de um conjunto de filtros de Kalman que são inicializados com valores distintos determinísticos de distância estimada do alvo. O autor ressalta que este método pode ser aplicado a qualquer variação de Filtro de Kalman, aplicando-o ao MP-EKF em seu trabalho.

De forma geral, o RP-EKF e suas variações podem ser vistos como uma forma sofisticada de inicializar o problema, dado que a distância que queremos calcular como variável de estado precisa ser estimada inicialmente. Sendo assim, é proposto que se defina uma faixa de valores para  $\hat{r}(0)$ ,  $\hat{r}(0) \in [\hat{r}_{\min}(0), \hat{r}_{\max}(0)]$ , segmentada em  $N_{i_r}$  intervalos (não necessariamente iguais<sup>4</sup>), com cada intervalo  $i_r$  cobrindo uma distância de  $\hat{r}_{i_r,\min}(0)$  a  $\hat{r}_{i_r,\max}(0)$ , como esquematizado na Figura 3.2.

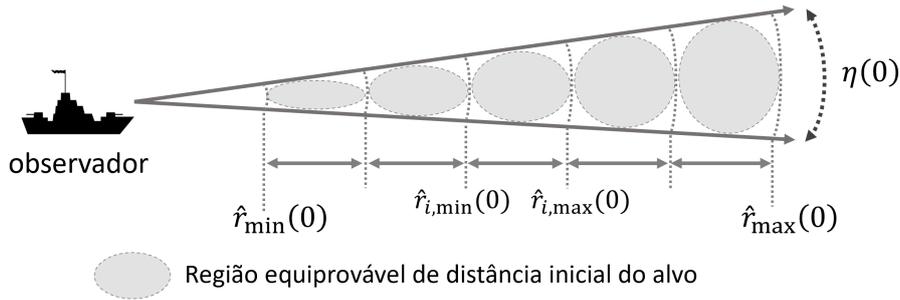


Figura 3.2: Diagrama de inicialização dos filtros

Em relação à inicialização da velocidade do alvo, PEACH [23] se baseia em [21], fazendo  $\hat{\mathbf{v}}_t(0) = \mathbf{v}_o(0)$  e, para o MP-EKF,  $\hat{\beta}(0) = \hat{\rho}(0) = 0$  (por este motivo o modelo é normalmente referenciado apenas como “Parametrização de Distância”). Já LA SCALA e MORELANDE [25] utilizam a divisão em intervalos também para estimar a velocidade inicial do alvo. Para a velocidade, os autores trabalham com o mesmo número de intervalos  $N_{i_v} = N_{i_r}$  e associam as menores velocidades às menores distâncias, mantendo, assim, a quantidade total de filtros calculados inalterada.

No entanto, é possível fazer uma busca mais ampla dos parâmetros iniciais combinando exaustivamente os intervalos de distância e velocidade, tendo um total de  $N_i = N_{i_r} \times N_{i_v}$  filtros. Aplica-se o mesmo raciocínio descrito anteriormente, definindo uma faixa de valores  $\hat{v}_t(0) \in [\hat{v}_{t,\min}(0), \hat{v}_{t,\max}(0)]$ , segmentada em  $N_{i_v}$  intervalos distintos, cobrindo, cada um, uma faixa de velocidade de  $\hat{v}_{t,i_v,\min}(0)$  a  $\hat{v}_{t,i_v,\max}(0)$ .

<sup>4</sup>Para cobrir uma faixa grande de distâncias, PEACH [23] utiliza intervalos que aumentam em progressão geométrica, de forma que as distâncias menores tenham menor variância e mais precisão na busca, mas também analisando distâncias maiores aceitando uma incerteza maior.

Sendo assim, cada filtro  $i$  é inicializado com uma combinação de valor de uma faixa  $i_r$  de distância e de uma faixa  $i_v$  de velocidade do alvo, com média e desvios-padrão dados por (3.85), calculando  $\hat{\mathbf{x}}_i(0|0)$  por (3.13) e a matriz de covariância  $\mathbf{P}_{e,i}(0|0)$  estimada por (3.14).

$$\begin{cases} \hat{r}_i(0) = \frac{1}{2} [\hat{r}_{i_r,\min}(0) + \hat{r}_{i_r,\max}(0)] \\ \sigma_{\hat{r},i}(0) = \frac{1}{\sqrt{12}} [\hat{r}_{i_r,\max}(0) - \hat{r}_{i_r,\min}(0)] \\ \hat{v}_{t,i}(0) = \frac{1}{2} [\hat{v}_{t,i_v,\min}(0) + \hat{v}_{t,i_v,\max}(0)] \\ \sigma_{\hat{v}_{t,i}}(0) = \frac{1}{\sqrt{12}} [\hat{v}_{t,i_v,\max}(0) - \hat{v}_{t,i_v,\min}(0)] \end{cases} \quad (3.85)$$

Os filtros são executados de forma independente um do outro, podendo ser processados em computação paralela. Os resultados individuais  $\hat{\mathbf{x}}_i(k|k)$  e  $\mathbf{P}_{e,i}(k|k)$  são ponderados por um fator  $\xi_i(k)$ , com  $\sum_{i=1}^{N_i} \xi_i(k) = 1$ , para uma resposta final, dada por [25]:

$$\hat{\mathbf{x}}(k|k) = \sum_{i=1}^{N_i} \xi_i(k) \hat{\mathbf{x}}_i(k|k) \quad (3.86)$$

e

$$\mathbf{P}_e(k|k) = \sum_{i=1}^{N_i} \xi_i(k) \left\{ \mathbf{P}_{e,i}(k|k) + [\hat{\mathbf{x}}_i(k|k) - \hat{\mathbf{x}}(k|k)] [\hat{\mathbf{x}}_i(k|k) - \hat{\mathbf{x}}(k|k)]^T \right\} . \quad (3.87)$$

Os pesos  $\xi_i(k)$  são atualizados a cada iteração com base no teorema de Bayes com a premissa de que os erros de medição e de estimativa são gaussianos [23], conforme (3.88).

$$\xi_i(k) = \frac{\xi_i(k-1) \frac{1}{\sqrt{2\pi [\sigma_\eta^2(k) + \sigma_{\hat{\beta},i}^2(k)]}} \exp \left\{ -\frac{1}{2} \frac{[\hat{\beta}_i(k|k) - \tilde{\beta}(k)]^2}{\sigma_\eta^2(k) + \sigma_{\hat{\beta},i}^2(k)} \right\}}{\sum_{i=1}^{N_i} \xi_i(k-1) \frac{1}{\sqrt{2\pi [\sigma_\eta^2(k) + \sigma_{\hat{\beta},i}^2(k)]}} \exp \left\{ -\frac{1}{2} \frac{[\hat{\beta}_i(k|k) - \tilde{\beta}(k)]^2}{\sigma_\eta^2(k) + \sigma_{\hat{\beta},i}^2(k)} \right\}} \quad (3.88)$$

Para coordenada cartesiana,  $\hat{\beta}_i(k|k)$  é dada por (3.89) e  $\sigma_{\hat{\beta},i}^2(k)$  é dada por (3.90), com  $\sigma_{\hat{r}_x}^2(k|k)$  e  $\sigma_{\hat{r}_y}^2(k|k)$  sendo estimadas pelas primeira e segunda componente da diagonal principal de  $\mathbf{P}_{e,i}(k|k)$ , respectivamente. Para polar modificada e log-polar,  $\hat{\beta}_i(k|k)$  é a primeira componente da variável de estado  $\hat{z}_i(k|k)$  e  $\sigma_{\hat{\beta},i}^2(k)$  estimada pela primeira componente da diagonal principal de  $\mathbf{P}_{e,MP,i}(k|k)$  ou  $\mathbf{P}_{e,LP,i}(k|k)$ , res-

pectivamente.

$$\hat{\beta}_i(k|k) = \arctan \frac{\hat{r}_{x,i}(k|k)}{\hat{r}_{y,i}(k|k)} \quad (3.89)$$

$$\sigma_{\hat{\beta}_i}^2(k) = \frac{\hat{r}_{y,i}^2(k|k)\sigma_{\hat{r}_x}^2(k|k) + \hat{r}_{x,i}^2(k|k)\sigma_{\hat{r}_y}^2(k|k)}{[\hat{r}_{x,i}^2(k|k) + \hat{r}_{y,i}^2(k|k)]^2} \quad (3.90)$$

Com base em (3.88), é visto que, quanto menor a diferença entre a marcação estimada e medida, maior será o peso do filtro, e que, em um bom cenário de rastreamento de alvo, tenderá a dominar a solução. Por conta disso, a fim de reduzir o processamento, é proposto em [23] que se estabeleça um limiar  $\xi_{\min}$  para que o filtro possa ser descartado e não mais recalculado, atribuindo peso zero e normalizando-os novamente para garantir a soma unitária. No entanto, o autor não define um tratamento para os filtros descartados, sugerindo que possam ser reiniciados com outros valores. Os valores iniciais  $\xi_i(0)$  podem ser definidos de forma uniforme ( $\xi_i(0) = \frac{1}{N_i}$ ) ou por alguma distribuição de probabilidade que se tenha conhecimento para a situação aplicada.

Um resumo do algoritmo pode ser visto na Tabela 3.10. Caso seja aplicada para um algoritmo em coordenada polar ou variação, basta converter as variáveis iniciais dos filtros.

Tabela 3.10: Filtro de Kalman Estendido com Parametrização de Distância e Velocidade

---

**RP-EKF**

---

$\hat{r}_{i_r, \min}(0)$ ,  $\hat{r}_{i_r, \max}(0)$ ,  $\hat{v}_{t, i_r, \min}(0)$  e  $\hat{v}_{t, i_r, \max}(0)$  definidos  
 $i = 0$

Para  $i_r$  de 1 a  $N_{i_r}$ :

Para  $i_v$  de 1 a  $N_{i_v}$ :

$$i = i + 1$$

$$\hat{r}_i(0) = \frac{1}{2} [\hat{r}_{i_r, \min}(0) + \hat{r}_{i_r, \max}(0)]$$

$$\sigma_{\hat{r}, i}(0) = \frac{1}{\sqrt{12}} [\hat{r}_{i_r, \max}(0) - \hat{r}_{i_r, \min}(0)]$$

$$\hat{v}_{t, i}(0) = \frac{1}{2} [\hat{v}_{t, i_v, \min}(0) + \hat{v}_{t, i_v, \max}(0)]$$

$$\sigma_{\hat{v}_{t, i}}(0) = \frac{1}{\sqrt{12}} [\hat{v}_{t, i_v, \max}(0) - \hat{v}_{t, i_v, \min}(0)]$$

Calcular  $\hat{\mathbf{x}}_i(0|0)$  e  $\mathbf{P}_{e, i}(0|0)$ , conforme (3.13) e (3.14)

(Fazer mudança de coordenada, se necessário)

$$\xi_i(0) = \frac{1}{N_i}$$

Para  $k > 0$ :

$$\Xi(k) = 0$$

Para  $i$  de 1 a  $N_i$ :

Se  $\xi_i(k-1) > 0$ :

Calcular  $\hat{\mathbf{x}}_i(k|k)$  e  $\mathbf{P}_{e, i}(k|k)$ , conforme *loop* do algoritmo escolhido

Calcular  $\hat{\beta}_i(k|k)$  e  $\sigma_{\hat{\beta}, i}^2(k)$ , conforme tipo de coordenada

$$\xi_i(k) = \xi_i(k-1) \frac{1}{\sqrt{2\pi [\sigma_{\eta}^2(k) + \sigma_{\hat{\beta}, i}^2(k)]}} \exp \left\{ -\frac{1}{2} \frac{[\hat{\beta}_i(k|k) - \tilde{\beta}(k)]^2}{\sigma_{\eta}^2(k) + \sigma_{\hat{\beta}, i}^2(k)} \right\}$$

$$\Xi(k) = \Xi(k) + \xi_i(k)$$

Senão:  $\xi_i(k) = 0$

$$\Xi_2(k) = 0$$

Para  $i$  de 1 a  $N_i$ :

$$\xi_i(k) = \frac{\xi_i(k)}{\Xi(k)}$$

Se  $\xi_i(k) \geq \xi_{\min}$ :  $\Xi_2(k) = \Xi_2(k) + \xi_i(k)$

Senão:  $\xi_i(k) = 0$

$$\text{Para } i \text{ de } 1 \text{ a } N_i: \xi_i(k) = \frac{\xi_i(k)}{\Xi_2(k)}$$

$$\hat{\mathbf{x}}(k|k) = \mathbf{0}; \quad \mathbf{P}_e(k|k) = \mathbf{0}$$

Para  $i$  de 1 a  $N_i$ :

Se  $\xi_i(k) > 0$ :  $\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k) + \xi_i(k) \hat{\mathbf{x}}_i(k|k)$

Para  $i$  de 1 a  $N_i$ :

Se  $\xi_i(k) > 0$ :

$$\mathbf{P}_e(k|k) = \mathbf{P}_e(k|k) + \xi_i(k) \left\{ \mathbf{P}_{e, i}(k|k) + [\hat{\mathbf{x}}_i(k|k) - \hat{\mathbf{x}}(k|k)] [\hat{\mathbf{x}}_i(k|k) - \hat{\mathbf{x}}(k|k)]^T \right\}$$


---

# Capítulo 4

## Preparação dos Experimentos

Os algoritmos descritos no capítulo anterior foram executados por meio de simulação computacional. A metodologia usada, os cenários simulados e as métricas de desempenho para comparação dos resultados serão apresentados nas próximas seções.

### 4.1 Metodologia

Após revisão bibliográfica dos trabalhos relacionados, identificando um histórico dos modelos e suas principais inovações e melhorias, assim como a acurácia da estimativa dos parâmetros e o tempo de processamento, os modelos EKF, PLKF, BC-PLKF, SAM-IVKF, EnKF, AEnKF (com polinômio de 2° grau), MP-EKF e LP-EKF foram programados em MATLAB. O RP-EKF foi considerado como uma forma de inicialização dos modelos, sendo, então, programados 8 modelos adicionais como versões dos algoritmos com parametrização de distância e velocidade (RP-EKF, RP-PLKF, RP-BC-PLKF, RP-SAM-IVKF, RP-EnKF, RP-AEnKF, RP-MP-EKF e RP-LP-EKF).

Foram aplicados os modelos na forma algébrica apresentada anteriormente, utilizando funções prontas do MATLAB quando disponíveis para economizar processamento (como média, covariância e Fatoração de Cholesky para gerar ruído a partir da matriz de covariância), além de criar outras mais específicas para algum modelo. A otimização dos produtos matriciais ou inversão de matrizes foi feita utilizando os operadores de inversão do MATLAB quando encadeado a um produto (“/” e “\”) ao invés da função “INV” para ganho de tempo de processamento. No entanto, não foi feita uma análise algébrica da possibilidade de otimização dos produtos matriciais e inversão de matrizes, o que também não foi encontrado nas referências levantadas.

Cada algoritmo foi implementado como uma função para o *loop* principal, de forma que todos fossem executados em paralelo para um mesmo cenário, sem que o resultado de um interferisse no outro e com os tempos de processamento computados

individualmente. Os algoritmos são chamados por um código principal, onde os cenários foram parametrizados. Todos os códigos encontram-se no Apêndice B.

Quanto à execução, cada cenário distinto de rota e manobra possui uma determinada duração, de  $k = 1$  a  $K$  amostras, e foram rodadas  $N = 2.000$  simulações de Monte Carlo em situações distintas de inicialização, descritas na Seção 4.2.

Ao fim do processamento, os resultados foram computados e as métricas de desempenho definidas na Seção 4.3 calculadas para comparação e análise de desempenho dos modelos.

## 4.2 Cenários Simulados

A definição dos cenários considera dois principais fatores: forma de inicialização e trajetória dos navios. Foram combinadas 5 formas de inicialização com 3 trajetórias dos navios, totalizando 15 cenários, que serão descritos nas próximas subseções. Os demais parâmetros dos modelos foram estabelecidos comuns a todos os cenários.

### 4.2.1 Inicialização

Com base nos trabalhos levantados, foram identificadas quatro principais formas de inicializar as variáveis do modelo, sendo a primeira uma abordagem mais teórica e a quarta uma variação da terceira. Além disso, foi adotada como quinta forma de inicialização a aplicação do modelo de parametrização de distância e velocidade.

- i. **Inicialização 1 ( $I_1$ ):** inicialização aleatória, com a estimativa  $\hat{\mathbf{x}}(0|0)$  como variável aleatória centrada no valor real e com desvio-padrão arbitrado, gerando um novo valor para cada simulação de Monte Carlo, como em [8] e [9];
- ii. **Inicialização 2 ( $I_2$ ):** inicialização aleatória, com a estimativa  $\hat{\mathbf{x}}(0|0)$  como variável aleatória com média e desvio-padrão arbitrados, gerando um novo valor para cada simulação de Monte Carlo, como em [7] e [25];
- iii. **Inicialização 3 ( $I_3$ ):** inicialização determinística e  $\hat{\mathbf{x}}(0|0)$  em um valor arbitrado, com a aleatoriedade apenas no erro de medida da marcação, sendo estimados os desvios-padrão das variáveis apenas para calcular  $\mathbf{P}(0|0)$ , como em [6] e [20];
- iv. **Inicialização 4 ( $I_4$ ):** similar a  $I_3$ , porém estimando apenas a distância  $\hat{\mathbf{r}}(0|0)$ , mantendo  $\hat{\mathbf{v}}(0|0) = \mathbf{0}$ , ou seja,  $\hat{\mathbf{v}}_t(0) = \mathbf{v}_o(0)$ , já que se desconhece o rumo do alvo (recomendado em [20], [21] e [23]). Para o caso particular do PLKF, será aplicada a solução trivial descrita na Seção 3.3, dado que este modelo permite esta inicialização; e

- v. **Inicialização 5 (I<sub>5</sub>)**: aplicação da parametrização de distância e velocidade, fazendo a combinação de 8 intervalos de distância com 8 intervalos de velocidade do alvo, num total de 64 filtros distintos.

Em [25] e [9], o rumo do alvo é um parâmetro estimado com distribuição uniforme no intervalo  $[-\frac{\pi}{2}, \frac{\pi}{2}]$  e o seu valor não foi sorteado, mas adotado  $\hat{\alpha}_t = \tilde{\beta}(0) + \pi$  (ou seja, o alvo com rumo de encontro ao observador). Foi utilizada, então, esta inicialização para I<sub>1</sub>, I<sub>2</sub>, I<sub>3</sub> e I<sub>5</sub>. Sendo assim, os valores aplicados para todos os cenários (média e desvio-padrão) encontram-se na Tabela 4.1.

Para a inicialização de  $\mathbf{P}_e(0|0)$ , foi adotada (3.14) para os modelos de coordenada cartesiana. Para o MP-EKF foi usada (3.73) nos cenários I<sub>1</sub>, I<sub>2</sub>, I<sub>3</sub> e I<sub>5</sub>, e (3.72) no cenário I<sub>4</sub>. Para o LP-EKF foi usada (3.82) nos três primeiros cenários e no quinto, e (3.84) no quarto.

Tabela 4.1: Parâmetros para inicialização por cenário

Cenário	Distância ao alvo (nmi)		Velocidade do alvo (nós)		Rumo do alvo (rad)	
	$\bar{r}(0 0)$	$\sigma_{\bar{r}}$	$\bar{v}_t(0 0)$	$\sigma_{\bar{v}_t}$	$\hat{\alpha}_t$	$\sigma_{\hat{\alpha}_t}$
I <sub>1</sub>	Conforme rota	2, 2	Conforme rota	4, 0	$\tilde{\beta}(0) + \pi$	$\frac{\pi}{\sqrt{12}}$
I <sub>2</sub> e I <sub>3</sub>	15	2, 2	15	4, 0	$\tilde{\beta}(0) + \pi$	$\frac{\pi}{\sqrt{12}}$
I <sub>4</sub> *	15	2, 2	$v_o(0)$	4, 0	$\alpha_o(0)$	$\frac{\pi}{\sqrt{12}}$
I <sub>5</sub>	De 2 a 20	$\frac{9}{4\sqrt{12}}$	De 5 a 20	$\frac{15}{8\sqrt{12}}$	$\tilde{\beta}(0) + \pi$	$\frac{\pi}{\sqrt{12}}$

\*Exceto PLKF, com  $\hat{\mathbf{x}}(0|0) = \mathbf{0}$  e  $\mathbf{P}_e(0|0) = \mathbf{I}$ .

## 4.2.2 Movimentação dos Navios

Foram escolhidos 3 cenários de movimentação de navios, sendo os 2 primeiros com 2 pernadas e o terceiro com 3 pernadas.

- i. **Movimentação 1 (M<sub>1</sub>)**: Navio-observador realiza duas pernadas com velocidades constantes e distintas, manobra com representação simplificada, alvo em MRU, sendo representada na Figura 4.1. Esta movimentação simplificada, usada em [6] e [9], considera que uma rotação de 3° por segundo permitiria uma guinada de até 180° dentro de um período de amostragem de 1 minuto (ou seja, em geral não haveria dados durante a manobra). A guinada é acentuada (134°) e, segundo SUN e XIN [9], gera uma maior descontinuidade e impõe

maior dificuldade para processamento. O rumo do alvo é facilitado, sendo igual à marcação inicial somada de  $180^\circ$  (alvo de encontro ao observador), que foi adotado em [25] e [9]. Os detalhes da movimentação são:

(a) Navio-observador

Parâmetros iniciais ( $t = 0$ ):  $s_o(0) = [0 \ 0]^T$ ,  $v_o(0) = 16$  nós,  $\alpha_o(0) = 280^\circ$

Manobra em  $t = 8$  min:  $v_o(8) = 10$  nós,  $\alpha_o(8) = 146^\circ$ , até  $t = 16$  min.

(b) Navio-alvo:

MRU:  $r(0) = 8$  nmi,  $\beta(0) = 225^\circ$ ,  $v_t = 12$  nós,  $\alpha_t = 45^\circ$ .

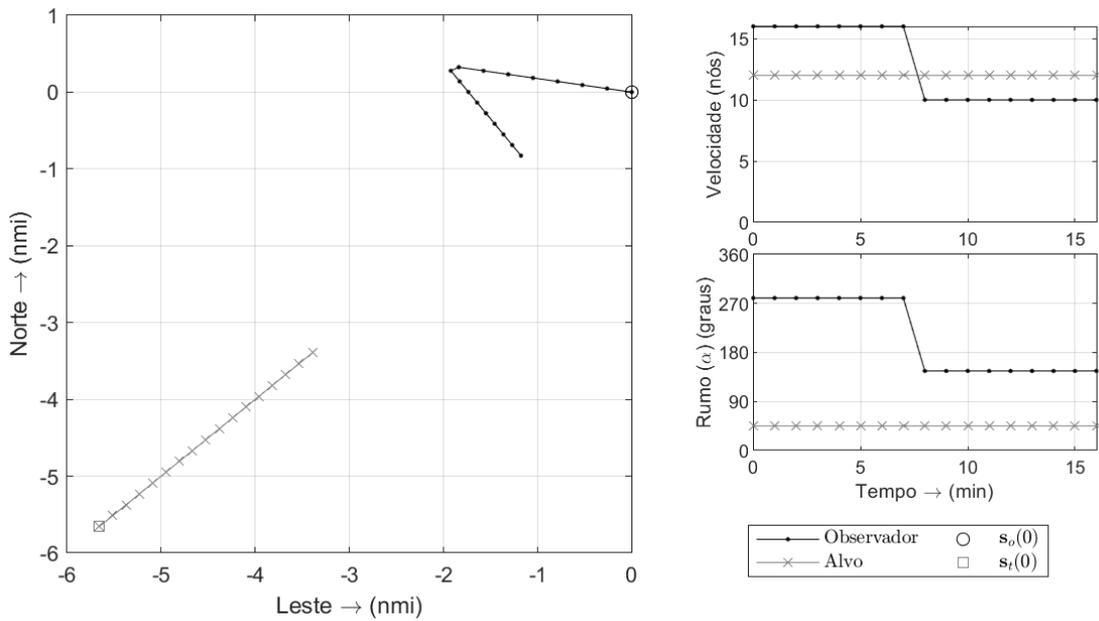


Figura 4.1: Movimentação 1

ii. **Movimentação 2 ( $M_2$ ):** Navio-observador realiza duas pernas com velocidades constantes e iguais, manobra mais detalhada, alvo em MRU, sendo representada na Figura 4.2. O desenho da curva de guinada é detalhado, aplicando conceitos de navegação, que permite que se haja amostragem durante a manobra, o que ocorrerá em uma situação real. A guinada manteve-se acentuada ( $105^\circ$ ) e não foi imposta condição para o rumo do alvo. Os detalhes da movimentação são:

(a) Navio-observador

Parâmetros iniciais ( $t = 0$ ):  $s_o(0) = [0 \ 0]^T$ ,  $v_o(0) = 12$  nós,  $\alpha_o(0) = 5^\circ$

Manobra iniciada em  $t = 10$  min mantendo a velocidade constante, com objetivo de guinada para  $\alpha_o = 110^\circ$ , o que gera um avanço final de 330

jardas<sup>1</sup> (0,1629 nmi) e um afastamento de 280 jardas (0,1382 nmi) [36], e continuando a segunda perna com  $v_o(t) = 12$  nmi até  $t = 20$  min.

(b) Navio-alvo:

MRU:  $r(0) = 10$  nmi,  $\beta(0) = 60^\circ$ ,  $v_t = 15$  nós,  $\alpha_t = 210^\circ$ .

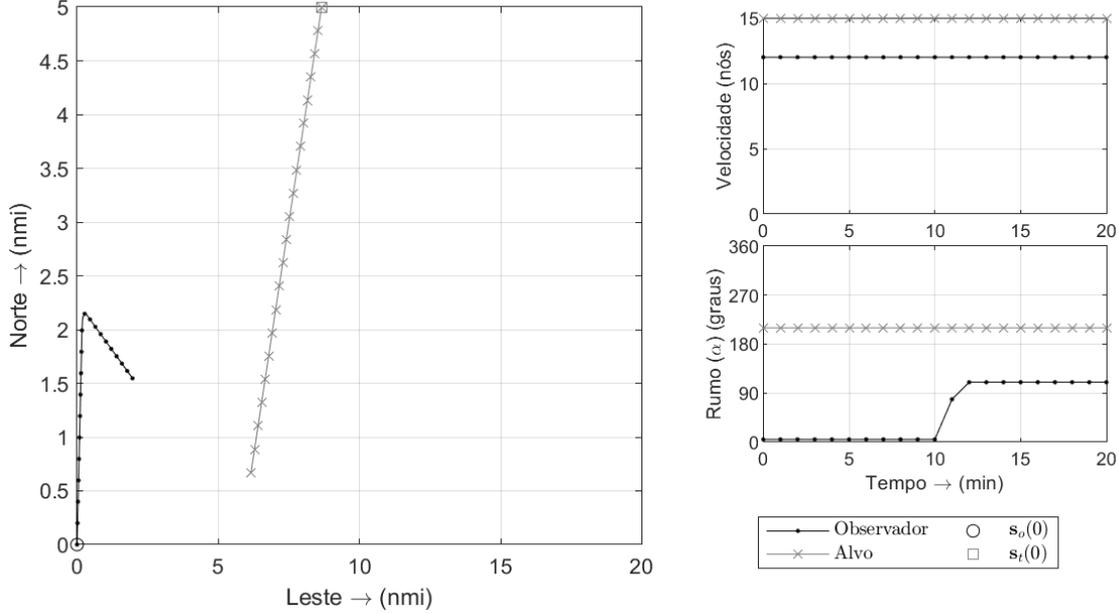


Figura 4.2: Movimentação 2

iii. **Movimentação 3 (M<sub>3</sub>):** Navio-observador realiza três pernas com velocidades constantes e distintas, manobras mais detalhadas, alvo em MRU, sendo representada na Figura 4.3. Seguindo o mesmo princípio de M<sub>2</sub>, neste cenário foi incluída uma terceira perna. A primeira guinada não é acentuada (60°), ainda assim acima dos 50° mínimos recomendados por SON *et al.* [18]. Já a segunda guinada é de 90°. Os detalhes da movimentação são:

(a) Navio-observador

Parâmetros iniciais ( $t = 0$ ):  $s_o(0) = [0 \ 0]^T$ ,  $v_o(0) = 15$  nós,  $\alpha_o(0) = 115^\circ$

Primeira manobra iniciada em  $t = 10$  min mantendo a velocidade constante, com objetivo de guinada para  $\alpha_o = 55^\circ$ , o que gera um avanço final de 387,5 jardas (0,1913 nmi) e um afastamento de 137,5 jardas (0,0679 nmi).

Segunda perna segue em MRU, com  $v_o(t) = 12$  nmi.

Segunda manobra iniciada em  $t = 18$  min mantendo a velocidade constante, com objetivo de guinada para  $\alpha_o = 325^\circ$ , o que gera um avanço

<sup>1</sup>1 jarda corresponde a 0,9144 metro, ou  $4,9374 \times 10^{-4}$  milhas náuticas.[36]

final de 335 jardas (0,1654 nmi) e um afastamento de 220 jardas (0,1086 nmi).

Terceira pernada segue em MRU, com  $v_o(t) = 13$  nmi até  $t = 28$  min.

(b) Navio-alvo:

MRU:  $r(0) = 12$  nmi,  $\beta(0) = 150^\circ$ ,  $v_t = 10$  nós,  $\alpha_t = 300^\circ$ .

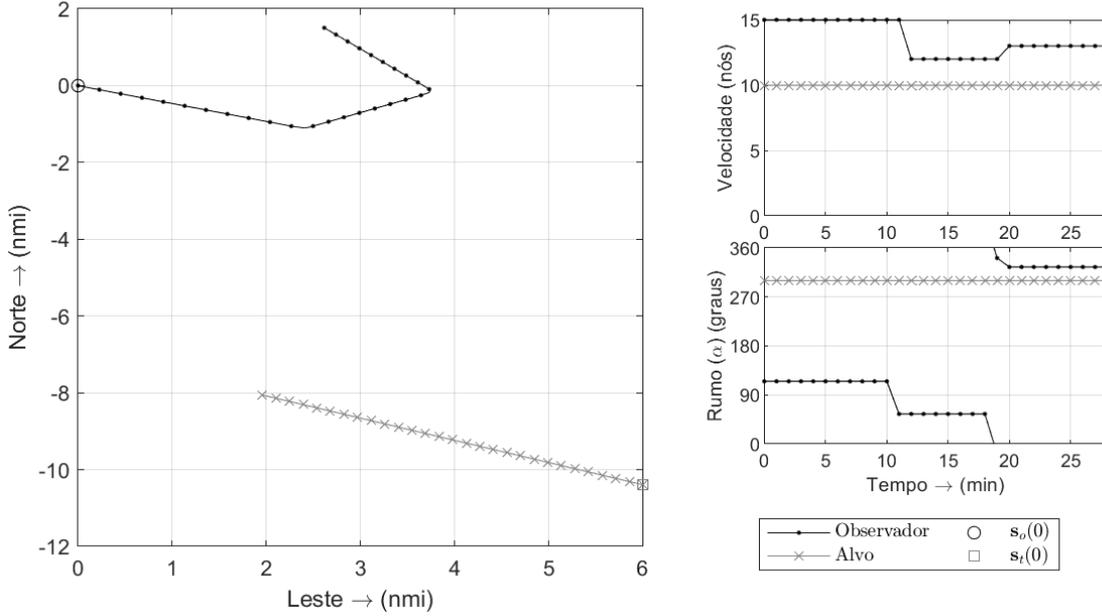


Figura 4.3: Movimentação 3

### 4.2.3 Demais Parâmetros

Os demais parâmetros adotados foram:

- i. comuns a todos os modelos e cenários:  $\sigma_\eta = 2^\circ$ ,  $\tau = 1$  min;
- ii. para o BC-PLKF:  $q_x = q_y = 0$ ;
- iii. para o SAM-IVKF:  $q_x = q_y = 0$ ,  $\kappa = 4$  [8]; e
- iv. para o EnKF e AEnKF:  $J = 500$ ,  $q_x = q_y = 9 \times 10^{-6} \text{ m}^2/\text{s}^3$  [9].

## 4.3 Métricas de Desempenho

Com base nos trabalhos levantados, foram escolhidas as principais métricas para comparação dos modelos. As métricas foram aplicadas para avaliação da distância ao alvo ( $r(k)$ ), velocidade do alvo ( $v_t(k)$ ) e rumo do alvo ( $\alpha_t$ ). Foram realizados

os cálculos instantâneos das métricas para estas variáveis, a fim de acompanhar a evolução do desempenho ao longo do tempo, simulando um acompanhamento em tempo real. Os dados do MP-EKF e LP-EKF foram convertidos para coordenada cartesiana, de forma a uniformizar a comparação.

Em [8] e [25] é sugerido ainda que, além dos valores instantâneos, as métricas sejam calculadas com uma média temporal, desconsiderando os primeiros instantes de *warm up* do processamento do modelo, onde o erro encontra-se elevado. Sendo assim, foi adotado como ponto inicial deste cálculo o instante ao final da manobra (ou da primeira manobra, se houver mais de uma), indicado por  $k_m$ .

Alguns trabalhos mais antigos, como [6] e [20], apresentam os resultados como o valor médio e desvio-padrão das variáveis medidas e, embora não seja mostrado nos trabalhos atuais, também será apresentado em momento oportuno.

### 4.3.1 Raiz-Quadrada do Erro Quadrático Médio

O cálculo da Raiz-Quadrada do Erro Quadrático Médio (RMSE – *Root-Mean-Squared Error*) é adotado pela maioria dos trabalhos para avaliação de desempenho das estimativas obtidas. Sendo  $\hat{\mathbf{x}}^{(n)}(k|k)$  o valor de  $\hat{\mathbf{x}}(k|k)$  na  $n$ -ésima simulação (índice válido também para suas componentes), o cálculo do RMSE pode ser feito da forma descrita de (4.1) a (4.6) [9].

$$\text{RMSE}_r(k) = \sqrt{\frac{1}{N} \sum_{n=1}^N \left\{ \left[ \hat{r}_x^{(n)}(k|k) - r_x(k) \right]^2 + \left[ \hat{r}_y^{(n)}(k|k) - r_y(k) \right]^2 \right\}} \quad (4.1)$$

$$\text{RMSE}_{v_t}(k) = \sqrt{\frac{1}{N} \sum_{n=1}^N \left[ \hat{v}_x^{(n)}(k|k) - v_x(k) \right]^2 + \left[ \hat{v}_y^{(n)}(k|k) - v_y(k) \right]^2} \quad (4.2)$$

$$\text{RMSE}_{\alpha_t}(k) = \sqrt{\frac{1}{N} \sum_{n=1}^N \left[ \arctan \frac{\hat{v}_x^{(n)}(k|k) + v_{o_x}(k)}{\hat{v}_y^{(n)}(k|k) + v_{o_y}(k)} - \arctan \frac{v_{t_x}(k)}{v_{t_y}(k)} \right]_{[-\pi, \pi]}^2} \quad (4.3)$$

$$\overline{\text{RMSE}}_r = \frac{1}{K - k_m + 1} \sum_{k=k_m}^K \text{RMSE}_r(k) \quad (4.4)$$

$$\overline{\text{RMSE}}_{v_t} = \frac{1}{K - k_m + 1} \sum_{k=k_m}^K \text{RMSE}_{v_t}(k) \quad (4.5)$$

$$\overline{\text{RMSE}}_{\alpha_t} = \frac{1}{K - k_m + 1} \sum_{k=k_m}^K \text{RMSE}_{\alpha_t}(k) \quad (4.6)$$

Cabe ressaltar que em (4.3) a diferença entre os arcos tangentes deve ser trazida para o intervalo  $[-\pi, \pi)$  para que uma eventual descontinuidade de ciclo não gere

um erro majorado.

### 4.3.2 Limite Inferior de Cramér-Rao Posterior

O Limite Inferior de Cramér-Rao (CRLB) é uma referência de mínimo MSE possível usada em modelos estatísticos para sistemas invariantes no tempo, dado pela inversa da matriz de informação de Fisher. Para o caso de sistemas variantes no tempo, como o estudado, foi desenvolvida uma versão onde o vetor de parâmetros tem que ser considerado aleatório, a fim de compreender a variação no tempo. A esta nova versão foi dado o nome de Limite Inferior de Cramér-Rao Posterior (PCRLB) [37].

Pela definição, sendo  $\mathcal{J}$  a matriz de informação de Fisher, o PCRLB é dado por:

$$\mathbb{E} \left\{ [\hat{\mathbf{x}}(k|k) - \mathbf{x}(k)] [\hat{\mathbf{x}}(k|k) - \mathbf{x}(k)]^T \right\} \geq \mathcal{J}^{-1}(k) \triangleq \text{PCRLB}_{\mathbf{x}}(k) . \quad (4.7)$$

O cálculo de  $\mathcal{J}(k)$  proposto em [37] é feito de forma iterativa e a solução apresentada em [8] aplicada ao problema de BOT é dada por:

$$\mathcal{J}(k) = [\mathbf{Q} + \mathbf{A}\mathcal{J}^{-1}(k-1)\mathbf{A}^T]^{-1} + \frac{1}{\sigma_\eta^2} \mathbb{E} [\boldsymbol{\zeta}(k)\boldsymbol{\zeta}^T(k)] , \quad (4.8)$$

onde

$$\boldsymbol{\zeta}(k) = \begin{bmatrix} \frac{r_y(k)}{r_x^2(k) + r_y^2(k)} & \frac{-r_x(k)}{r_x^2(k) + r_y^2(k)} & 0 & 0 \end{bmatrix}^T . \quad (4.9)$$

e o valor esperado em (4.8) é calculado por simulação de Monte Carlo utilizando a estimativa de desvio-padrão para a distância ao alvo sobre o valor real (foram executadas 500 repetições para cada  $k$ ). Em [24] é proposto inicializar  $\mathcal{J}^{-1}(0) \approx \mathbb{E} \left\{ [\hat{\mathbf{x}}(0|0) - \mathbf{x}(0)] [\hat{\mathbf{x}}(0|0) - \mathbf{x}(0)]^T \right\} = \mathbf{P}_e(0|0)$ .

Com a matriz do PCRLB calculada, a diagonal principal contém os limites de MSE para as componentes de  $\mathbf{x}(k)$ . Sendo  $d_{\text{CR},i}(k)$  o  $i$ -ésimo elemento da diagonal principal, temos o cálculo do PCRLB como referencial para distância e velocidade do alvo definido de (4.10) a (4.13).

$$\text{PCRLB}_r(k) = \sqrt{d_{\text{CR},1}(k) + d_{\text{CR},2}(k)} \quad (4.10)$$

$$\text{PCRLB}_{v_t}(k) = \sqrt{d_{\text{CR},3}(k) + d_{\text{CR},4}(k)} \quad (4.11)$$

$$\overline{\text{PCRLB}}_r = \frac{1}{K - k_m + 1} \sum_{k=k_m}^K \text{PCRLB}_r(k) \quad (4.12)$$

$$\overline{\text{PCRLB}}_{v_t} = \frac{1}{K - k_m + 1} \sum_{k=k_m}^K \text{PCRLB}_{v_t}(k) \quad (4.13)$$

Desta forma, é possível estabelecer a referência para as métricas de RMSE para distância e velocidade do alvo. No entanto, não foi encontrada referência para o PCRLB do rumo do alvo.

### 4.3.3 Erro de Estimação Quadrático Normalizado Médio

O Erro de Estimação Quadrático Normalizado Médio (ANEES – *Average Normalized Estimation Error Squared*) é um método para avaliação de credibilidade, sendo uma versão da distância de Mahalanobis incluindo a média para remover a dependência da dimensão de  $\mathbf{x}(k)$  e também reduzir incertezas. Quanto mais próximo de 1 for o ANEES, mais confiável é o funcionamento do filtro; se for muito maior que 1 indica que o filtro funciona de forma muito otimista (ou pessimista, para muito menor que 1) em relação à estimativa de  $\mathbf{P}_e(k|k)$  [38]. O cálculo para o problema de BOT, onde  $\mathbf{x}(k) \in \mathbb{R}^4$ , é dado por [9]:

$$\begin{aligned} \text{ANEES}_{\mathbf{x}}(k) &= \frac{1}{N} \sum_{n=1}^N \frac{1}{4} \|\hat{\mathbf{x}}^{(n)}(k|k) - \mathbf{x}(k)\|_{[\mathbf{P}_e^{(n)}(k|k)]}^2 \\ &= \frac{1}{4N} \sum_{n=1}^N [\hat{\mathbf{x}}^{(n)}(k|k) - \mathbf{x}(k)]^T [\mathbf{P}_e^{(n)}(k|k)]^{-1} [\hat{\mathbf{x}}^{(n)}(k|k) - \mathbf{x}(k)] \end{aligned} \quad (4.14)$$

e a média temporal é dada por:

$$\overline{\text{ANEES}}_{\mathbf{x}} = \frac{1}{K - k_m + 1} \sum_{k=k_m}^K \text{ANEES}_{\mathbf{x}}(k) . \quad (4.15)$$

### 4.3.4 Tempo de Processamento

O cálculo do tempo de processamento total (TPT – *Total Processing Time*) considera o tempo total dos *loops* de cada algoritmo (LPT – *Loop Processing Time*). O valor médio do TPT e a média do LPT serão apresentados para efeito de ordem de grandeza, sabendo-se que variam de acordo com o *hardware* utilizado e configurações dos *softwares* executados. Por esta razão, foi calculado também o tempo relativo (RPT – *Relative Processing Time*) de cada algoritmo, utilizado o EKF como base de comparação. Estão desconsiderados os tempos de inicialização, comum a todos os algoritmos. Serão computados todos os instantes, de  $k = 1$  a  $k = K$ , sendo  $K$  variável de acordo com o tempo de simulação de cada cenário de movimentação.

$$\overline{\text{TPT}}_{\text{algoritmo}} = \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K \text{LPT}_{\text{algoritmo}}^{(n)}(k) \quad (4.16)$$

$$\overline{\text{LTP}}_{\text{algoritmo}} = \frac{1}{K} \overline{\text{TPT}}_{\text{algoritmo}} \quad (4.17)$$

$$\overline{\text{RPT}}_{\text{algoritmo}} = \frac{\overline{\text{TPT}}_{\text{algoritmo}}}{\overline{\text{TPT}}_{\text{EKF}}} \quad (4.18)$$

### 4.3.5 Estabilidade e Convergência

A estabilidade do Filtro de Kalman depende da matriz de covariância do erro  $\mathbf{P}_e(k|k)$ , que, se mal condicionada, pode provocar um colapso prematuro e não haver a convergência do resultado [21]. Para avaliar o condicionamento, foi escolhido o recíproco do número de condicionamento de norma 1 ( $\iota_1$ ), calculado pela função “RCOND” do MATLAB, onde quanto mais próximo de 0, pior o condicionamento da matriz, e quanto mais próximo de 1, mais bem condicionada a matriz.

O recíproco do número de condicionamento é dado por [39]:

$$\iota_1(k) = \frac{1}{\|\mathbf{P}_e(k|k)\|_1 \|\mathbf{P}_e^{-1}(k|k)\|_1} \quad (4.19)$$

e, sendo  $p_{e,ij}$  o elemento da linha  $i$  e coluna  $j$  de  $\mathbf{P}_e(k|k)$ , a norma 1 é dada por:

$$\|\mathbf{P}_e(k|k)\|_1 = \max_{1 \leq j \leq 4} \sum_{i=1}^4 |p_{e,ij}| . \quad (4.20)$$

Sendo assim, foi arbitrado o limiar  $\iota_{1,\min} = 10^{-12}$  de forma conservadora e definida a métrica de Taxa de Estabilidade (SR – *Stability Rate*) como:

$$\text{SR}_{\text{algoritmo}} = 1 - \frac{\text{número de simulações com } \iota_1(k) < \iota_{1,\min} \text{ em algum instante } k}{N} . \quad (4.21)$$

O colapso da matriz de covariância ao longo do processamento pode ser um fator crítico porque pode gerar denominadores extremamente pequenos, próximos a zero, nas equações do modelo, inviabilizando a continuidade do processamento do cálculo. Por este motivo, as rodadas de simulação onde  $\iota_1 < \iota_{1,\min}$  foram desconsideradas do cálculo das métricas RMSE, ANEES e PCRLB.

No entanto, é possível que o condicionamento esteja acima do limiar estabelecido e os resultados (em particular de distância) apresentem erros elevados. Baseado em [25], é definido que a rota convergiu quando a raiz do erro quadrático de  $r(k)$  ( $\text{RSE}_r$  – *Root-Squared Error*), dada por (4.22), é inferior a um limite estabelecido.

$$\text{RSE}_r(k) = \sqrt{\left[\hat{r}_x^{(n)}(k|k) - r_x(k)\right]^2 + \left[\hat{r}_y^{(n)}(k|k) - r_y(k)\right]^2} \quad (4.22)$$

Considerando que a ordem de grandeza das distâncias utilizadas nos cenários da

Seção 4.2 é de  $10^1$  nmi, foi definida a Taxa de Convergência de Rota (TCR – *Track Convergence Rate*) como sendo o percentual do número de simulações onde a média dos três últimos instantes  $k$  simulados ( $K - 2$  a  $K$ ) não ultrapasse a mesma ordem de grandeza das distâncias reais (ou seja, menor ou igual a  $10^{\frac{3}{2}} \approx 31.6$  nmi). Logo:

$$\text{TCR}_{\text{algoritmo}} = \frac{\text{número de simulações com } \sum_{k=K-2}^K \frac{1}{3} RSE_r(k) \leq 31.6}{N} . \quad (4.23)$$

# Capítulo 5

## Resultados e Discussão

Os 15 cenários foram processados para os 8 modelos, com o cômputo das métricas estabelecidas no capítulo anterior. Os resultados gerados encontram-se no Apêndice C e serão referenciados ou trazidos neste capítulo com um recorte diferente para a discussão.

### 5.1 Desempenho Geral dos Modelos

Com base nos resultados, não foi observado um único modelo dominante, havendo variação dos modelos que obtiveram os melhores  $\overline{\text{RMSE}}_r$  ou  $\overline{\text{RMSE}}_{v_t}$  conforme a inicialização ou manobra. Além disso, de forma geral a estimativa da distância ao alvo possui menor erro quando comparado à estimativa da velocidade do alvo (tanto módulo quanto rumo), o que é observado nos gráficos de RMSE. Este efeito foi relatado por HOELZER *et al.* [7] para o MP-EKF, mas se constata que ocorre para os demais modelos. Essa diferença é percebida quando se calcula a razão entre o PCRLB, que estima o RMSE mínimo teórico, e o valor real da grandeza, que, em média, é superior para a velocidade, visto na Tabela 5.1, indicando que o Filtro de Kalman aplicado a BOT com esta configuração de espaço de estados e variáveis possui esta limitação, independente do modelo adotado.

Tabela 5.1: Razão média entre PCRLB e valores reais por cenário de movimentação

	$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_3$
$\text{PCRLB}_r(k)/r(k)$	0,067	0,237	0,211
$\text{PCRLB}_{v_t}(k)/v_t(k)$	0,203	0,226	0,423

Obs.: A média temporal considerou  $k$  de  $k_m$  a  $K$

Os modelos baseados em coordenada polar (MP-EKF e LP-EKF) foram os que tiveram pior desempenho, destoando do desempenho dos demais, como pode ser visto nas Figuras C.2 e C.8 e na Tabela C.3, além de apresentarem baixa TCR,

chegando a 0% nos cenários com  $M_3$  (Tabelas C.12 a C.15) por problemas de estabilidade (não ocorrido em outros algoritmos). Embora o LP-EKF tenha obtido pontualmente o segundo melhor desempenho de  $\overline{\text{RMSE}}_r$  (mas não de  $\overline{\text{RMSE}}_{v_t}$ ) no cenário  $M_2I_5$  (Tabela C.10) e o segundo melhor desempenho de  $\overline{\text{RMSE}}_{v_t}$  (mas não de  $\overline{\text{RMSE}}_r$ ) e, ainda assim, 8 vezes maior que o  $\overline{\text{PCRLB}}_{v_t}$  no cenário  $M_1I_4$  (Tabela C.4), a inconstância de desempenho demonstra uma fragilidade destes modelos baseados em coordenada polar para as situações estudadas e, por isso, não serão mais considerados nas análises que se seguem.

O AEnKF também apresentou problemas de convergência nos cenários com  $M_3$ , com índices abaixo de 20% para as 4 primeiras inicializações (C.11 a C.14). Pelas Figuras C.11 a C.14, é observado que a divergência do RMSE ocorre durante ou logo após a primeira manobra, não sendo, então, um problema oriundo do fato de haver 2 manobras nestes cenários. O modelo também acabou não atingindo 100% de convergência com a inicialização  $I_1$  nas demais movimentações. Ainda assim, o AEnKF esteve entre os dois melhores resultados de  $\overline{\text{RMSE}}_{v_t}$  nos cenários  $M_1I_2$ ,  $M_1I_3$ ,  $M_1I_5$ ,  $M_2I_2$ ,  $M_2I_3$  e  $M_2I_5$  (Tabelas C.2, C.3, C.5, C.7, C.8 e C.10). No entanto, como visto nas Figuras C.5, C.7, C.8 e C.10, há uma redução da credibilidade dos resultados do modelo em relação aos demais com o aumento do  $\text{ANEES}(k)$  ao longo do tempo. Foi testado também o AEnKF com polinômio de 3º grau, mas sem melhoria nos resultados e optou-se por não o apresentar neste trabalho.

O PLKF manteve a estabilidade e convergência nos cenários simulados, como previsto por AIDALA [6]. No entanto, de forma geral, manteve um resultado de RMSE mais elevado que os demais de coordenada cartesiana, como observado nas Figuras C.1 a C.15. Atingiu o melhor resultado de  $\overline{\text{RMSE}}_r$  e  $\overline{\text{RMSE}}_{v_t}$  para  $M_1I_4$  e  $M_1I_5$ , sendo em  $I_4$  aplicada inicialização trivial específica deste modelo. Porém, o bom resultado não se repetiu para as demais movimentações com estas mesmas inicializações.

O EKF manteve um bom desempenho geral, considerando a simplicidade do algoritmo em relação aos demais, mantendo a convergência em 100% para os cenários de  $M_1$  e  $M_2$ , mas reduzindo a menos da metade em  $M_3$ , com as mesmas características observadas no AEnKF. Ainda obteve melhor resultado para  $\overline{\text{RMSE}}_r$  em 2 cenários ( $M_2I_2$  e  $M_2I_4$ , conforme Tabelas C.7 e C.9).

O EnKF, o BC-PLKF e o SAM-IVKF compõem a lista de modelos com os melhores resultados de RMSE. Em diversos cenários, como observado nas Figuras C.8 e C.12, as curvas de RMSE destes modelos chegam a se confundir. Na Tabela 5.2 pode ser visto que estes modelos obtiveram os melhores resultados nas principais métricas de desempenho.

O EnKF cumpriu seu papel de melhorar a convergência do EKF, atingindo 100% em todos os cenários. Ainda foi um dos modelos com melhor desempenho, estando

Tabela 5.2: Número de cenários por *ranking* das métricas

	$\overline{\text{RMSE}}_r$			$\overline{\text{RMSE}}_{v_t}$			$\overline{\text{RMSE}}_{\alpha_t}$			$\overline{\text{ANEES}}$		
	1º	2º	3º	1º	2º	3º	1º	2º	3º	1º	2º	3º
EKF	2	2	-	-	2	4	-	3	2	1	2	2
PLKF	3	1	1	2	1	1	1	-	-	-	-	2
BC-PLKF	-	3	10	2	1	6	2	5	3	4	5	5
SAM-IVKF	6	2	2	5	3	2	4	3	4	7	1	3
EnKF	3	5	2	2	5	1	4	1	4	3	4	3
AEnKF ( $P^{(2)}$ )	1	1	-	4	2	-	3	1	1	-	3	-
MP-EKF	-	-	-	-	-	1	1	-	1	-	-	-
LP-EKF	-	1	-	-	1	-	-	2	-	-	-	-

em 8 cenários entre os 2 menores  $\overline{\text{RMSE}}_r$  e em 7 cenários entre os 2 menores  $\overline{\text{RMSE}}_{v_t}$ . Também se destacou na credibilidade, com  $\overline{\text{ANEES}}$  mais próximo de 1 nos cenários  $M_2I_2$ ,  $M_3I_1$  e  $M_3I_2$  (Tabelas C.7, C.11 e C.12).

Em relação ao BC-PLKF, a sua TCR foi de praticamente 100% para todos os cenários. Em apenas uma execução no cenário  $M_2I_1$  do total de 30.000 realizadas não houve convergência, onde não foi identificada uma causa específica que não fosse a própria aleatoriedade da simulação. Embora não tenha obtido o melhor  $\overline{\text{RMSE}}_r$  em nenhum cenário, o BC-PLKF se mostrou consistente ficando entre os três melhores em 10 dos 15 cenários. Mesmo em terceiro lugar, os valores foram sempre muito próximos do segundo colocado, como na Figura C.13. Para o  $\overline{\text{RMSE}}_{v_t}$ , além de melhor desempenho nos cenários  $M_1I_1$  e  $M_2I_4$  (Tabelas C.1 e C.9), o modelo ficou entre os três melhores em mais 7 cenários. A credibilidade também foi elevada, ficando entre os três resultados mais próximo de 1 em 14 dos 15 cenários.

Já o SAM-IVKF obteve TCR de 100% em 10 cenários, e, nos 5 restantes, 4 tiveram resultado acima de 99%. Foi observado nestes cenários que, na maioria dos casos, o recálculo de  $\mathbf{P}_e(k|k)$  utilizando  $\hat{\mathbf{c}}_{IV}(k)$  (3.33) e  $\mathbf{g}_{IV}(k)$  (3.32), que são uma solução sub-ótima para descorrelacionar  $\hat{\mathbf{c}}(k)$  do ruído de medição, gerou uma matriz de covariância com condicionamento ruim (mas acima do limite da estabilidade) e elementos negativos na diagonal principal, o que desencadeou a divergência do filtro. Foi testado restringir a seleção de ângulos, reduzindo  $\kappa$  de 4 para 2, o que não evitou o problema. Isso acarretou pequena variação no TCR, ocorrendo, por exemplo, nos cenários  $M_1I_1$  (Tabela C.1) e  $M_2I_1$  (Tabela C.6) um pequeno aumento, de 99,1 % e 96,10% para 99,40% e 96,20%, respectivamente, e no cenário  $M_2I_2$  (Tabela C.7) ligeira redução de 99,60% para 99,55%.

Como em quatro dos cenários do SAM-IVKF com menor TCR a inicialização foi  $I_1$  ou  $I_4$ , que não serão de interesse prático (o que será discutido na Seção 5.2), o

modelo segue como um bom candidato, pois obteve melhor desempenho simultâneo para  $\overline{\text{RMSE}}_r$ ,  $\overline{\text{RMSE}}_{v_t}$  e  $\overline{\text{ANEES}}$  em 3 cenários ( $M_2I_3$ ,  $M_3I_3$  e  $M_3I_4$  - Tabelas C.8, C.13 e C.14), sendo, no total, 10 cenários com  $\overline{\text{RMSE}}_r$  e  $\overline{\text{RMSE}}_{v_t}$  entre os três menores. Em relação à credibilidade, apesar de ter  $\overline{\text{ANEES}}$  entre os três melhores em 11 de 15 cenários, as Figuras C.7 e C.12, dos cenários  $M_2I_2$  e  $M_3I_2$ , mostram uma tendência de otimismo na estimativa de  $\mathbf{P}_e(k|k)$  ao final da simulação que, no caso específico do primeiro cenário, pode ter levado ao erro relatado, acendendo um ponto de atenção para o emprego prático do modelo.

As Figuras 5.1, 5.3 e 5.5 representam o diagrama de movimentação dos navios e a localização estimada (posição média e execução única) de cenários onde os modelos BC-PLKF, SAM-IVKF e EnKF, respectivamente, obtiveram bons resultados. Para os mesmos cenários e modelos, nas Figuras 5.2, 5.4 e 5.6 são apresentados os valores médios de  $\hat{r}(t)$  e  $\hat{v}_t(t)$  e seus respectivos desvios-padrão, além do valor real. Sendo assim, as próximas análises terão maior foco nestes três modelos.

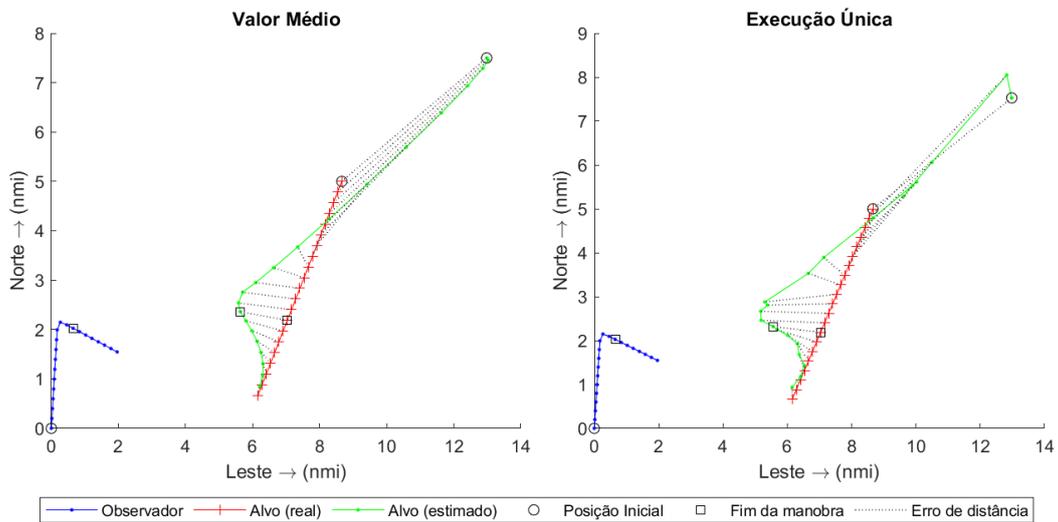


Figura 5.1: BC-PLKF –  $M_2I_4$  – Rastreamento

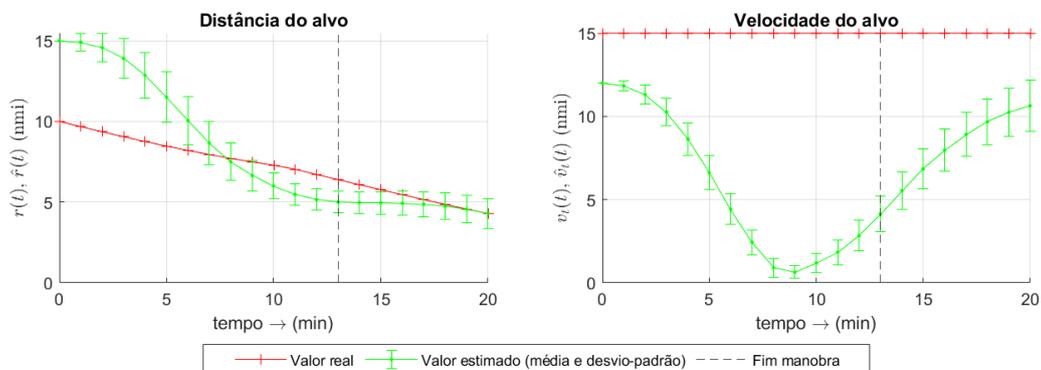


Figura 5.2: BC-PLKF –  $M_2I_4$  – Distância e Velocidade do alvo

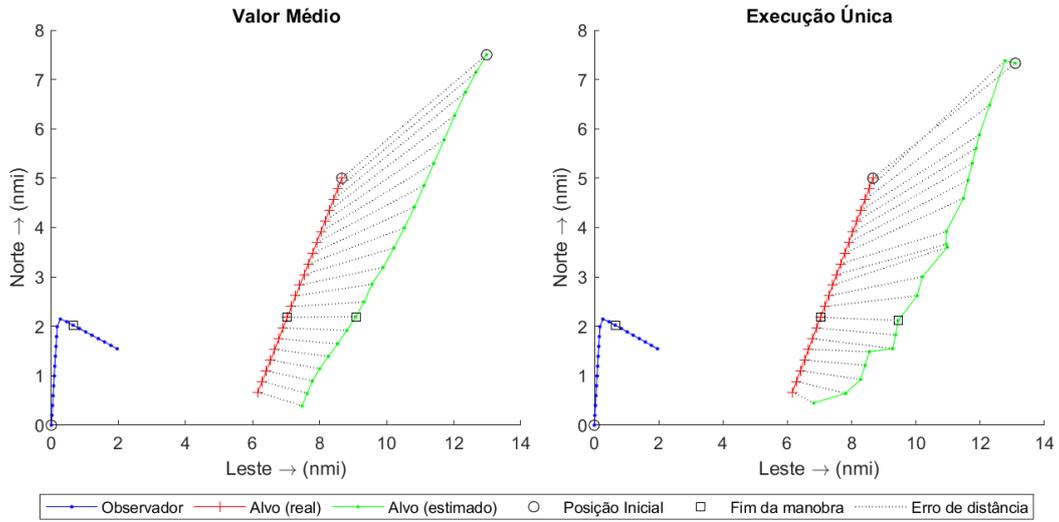


Figura 5.3: SAM-IVKF –  $M_2I_3$  – Rastreamento

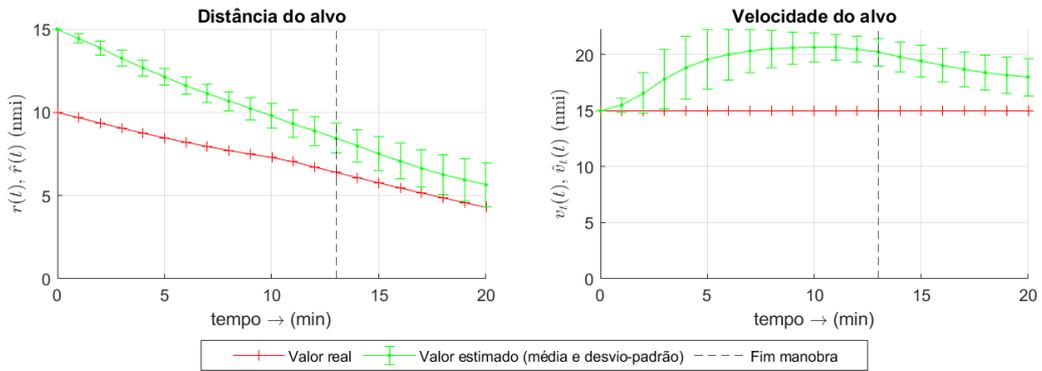


Figura 5.4: SAM-IVKF –  $M_2I_3$  – Distância e Velocidade do alvo

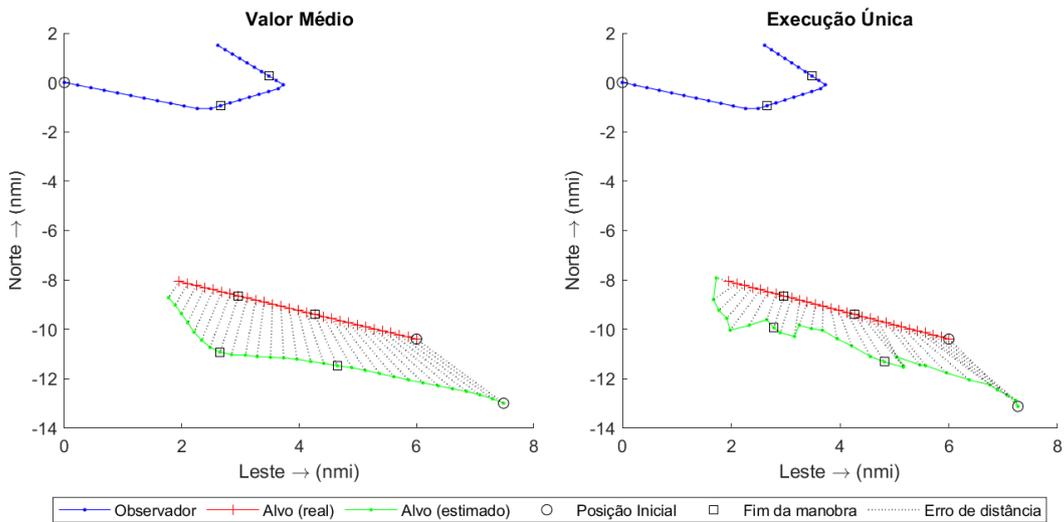


Figura 5.5: EnKF –  $M_3I_3$  – Rastreamento

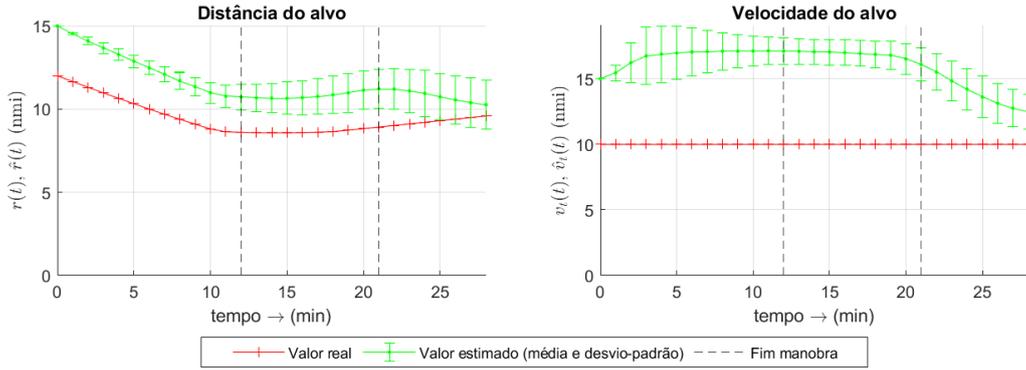


Figura 5.6: EnKF –  $M_3I_3$  – Distância e Velocidade do alvo

## 5.2 Impacto da Inicialização

Dentre os 5 tipos de inicialização, a  $I_1$  corresponde a um cenário teórico, sendo uma inicialização mais facilitada por ser centrada no valor real de distância e velocidade do alvo, enquanto  $I_4$  seria a situação mais dificultada, pois só parte de uma distância estimada (exceto para o PLKF, que foi pensado para iniciar com uma solução trivial). As inicializações  $I_2$  e  $I_3$  se aproximam de situações factíveis em aplicações práticas, assim como  $I_5$ , que tem alto potencial de uso prático.

Foi observado que os cenários com  $I_1$ , em especial o  $M_1I_1$ , tiveram uma baixa credibilidade com elevação do ANEES. Na Figura C.1 é visível que, mesmo em modelos como o EnKF com  $RMSE_r$  baixo, o ANEES foi elevado, indicando que a “facilitação excessiva” da inicialização pode levar a uma subestimação da matriz de covariância.

Nas Figuras 5.7, 5.8 e 5.9 são mostradas as métricas do BC-PLKF para  $M_1$ ,  $M_2$  e  $M_3$ , respectivamente, com o resultado das 5 inicializações no mesmo gráfico. Nesta visão fica claro o efeito de  $I_1$ , com melhor resultado de  $RMSE_r$  em alguns cenários, mas sempre destoando das demais métricas, em especial o ANEES, onde algumas vezes não é exibido nesta escala mais restrita adotada. Já a  $I_4$  só apresenta bom resultado em  $M_2$ , tanto para o BC-PLKF quanto para o SAM-IVKF, mas, ainda assim, com ANEES em desconformidade com os demais cenários. O mesmo resultado é visto para o SAM-IVKF e EnKF, que pode ser observado na Figura C.9 comparada com C.6, C.7, C.8 e C.10.

Com relação às demais inicializações, nota-se uma grande proximidade entre os resultados, com  $I_5$  tendo um desempenho um pouco inferior em  $RMSE_r$  em  $M_1$ , mas garantindo o melhor ANEES com certo destaque em todos os cenários, como visto nas Figuras 5.7 e 5.9. Comparando a Figura 5.7 com as Figuras 5.10 e 5.11, é possível notar o impacto da inicialização nos resultados, onde no cenário  $M_1I_1$  há uma convergência em média, reduzindo RMSE, mas com alto desvio-padrão, o que pode levar à piora do ANEES, ao passo que em  $M_1I_5$  os  $RMSE_r$  e  $RMSE_{v_t}$  não

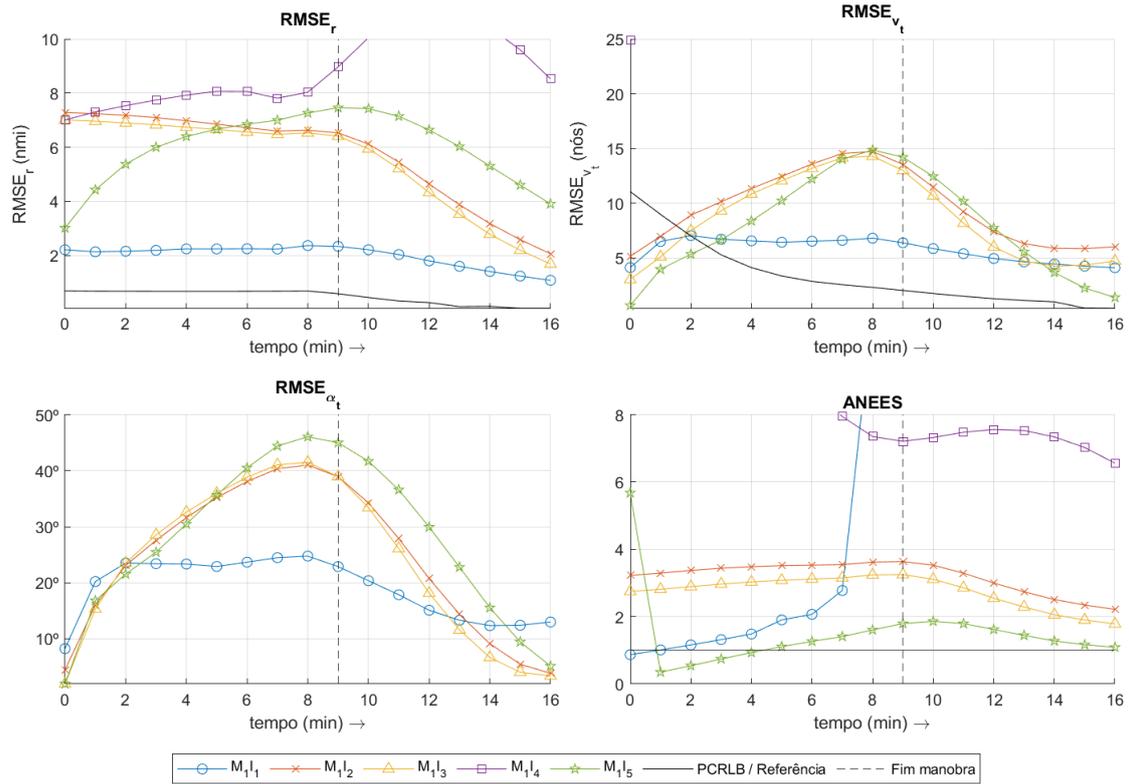


Figura 5.7: BC-PLKF –  $M_1$  – Métricas

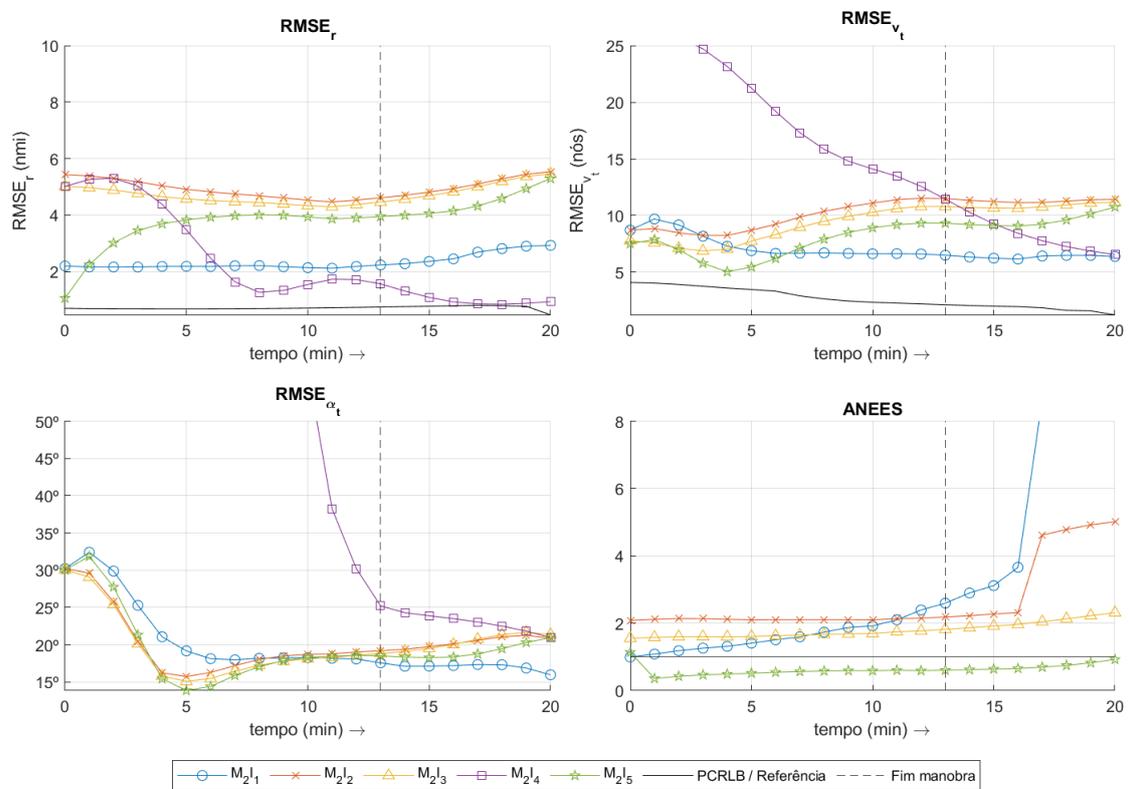


Figura 5.8: BC-PLKF –  $M_2$  – Métricas

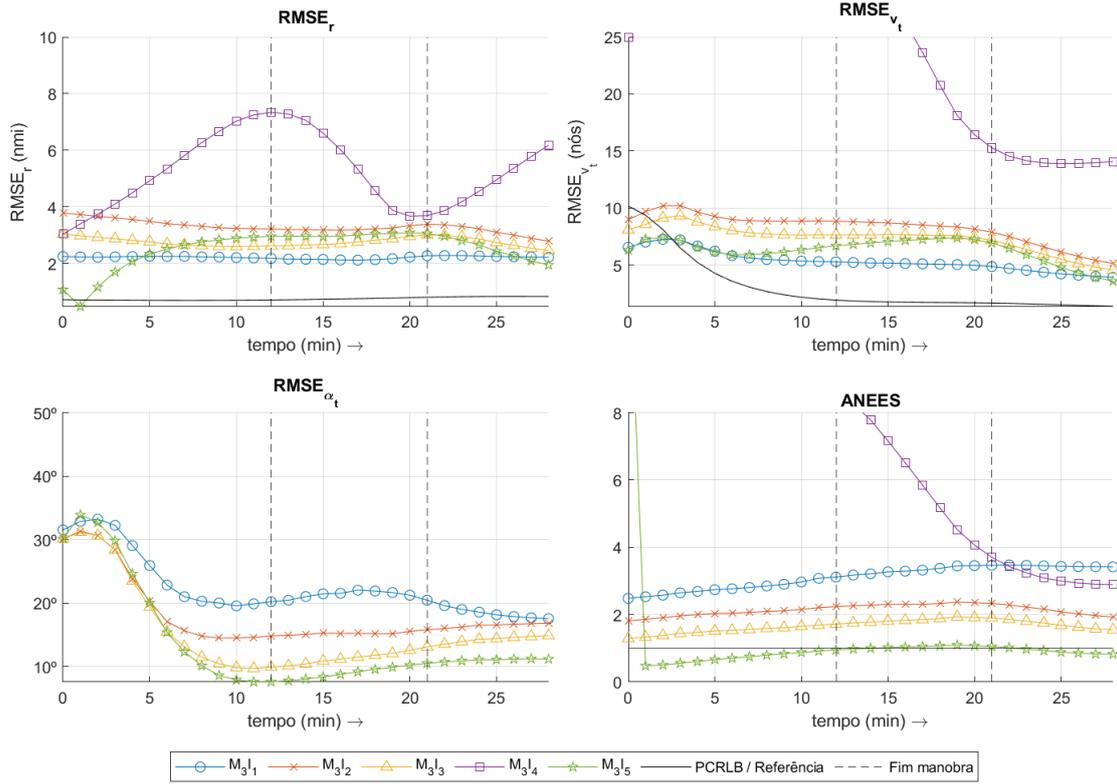


Figura 5.9: BC-PLKF –  $M_3$  – Métricas

tem o mesmo desempenho que no cenário anterior, mas possui baixo desvio-padrão, conferindo ao resultado maior credibilidade.

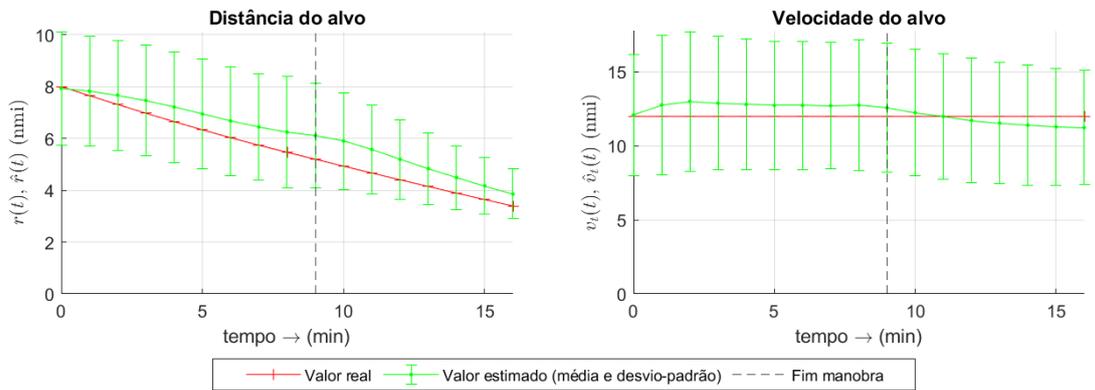


Figura 5.10: BC-PLKF –  $M_1I_1$  – Distância e Velocidade do alvo

Em vista disso, o uso das variações dos modelos com parametrização de distância e velocidade mostra-se efetivo nos cenários de movimentação mais realistas ( $M_2$  e  $M_3$ ), o que será abordado na próxima seção.

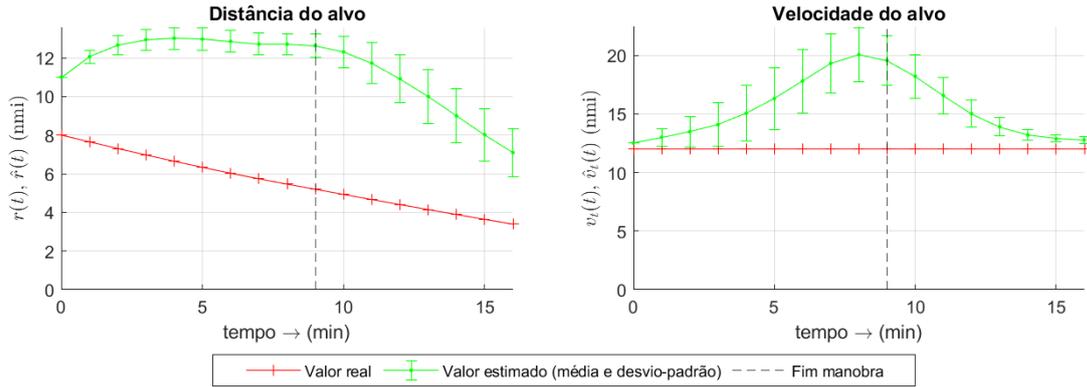


Figura 5.11: BC-PLKF –  $M_1I_5$  – Distância e Velocidade do alvo

### 5.3 Impacto da Movimentação dos Navios

Conforme descrito na Subseção 4.2.2, as movimentações foram definidas com base em referências adotadas em outros trabalhos e também buscando se aproximar de situações reais para cobrir algumas situações distintas, apesar deste trabalho não ter o intuito de definir uma melhor estratégia de manobra para o problema de BOT.

A movimentação  $M_1$  é uma representação simplificada com dados que facilitam a estimação do rumo do alvo para os cenários  $I_1$ ,  $I_2$ ,  $I_3$  e  $I_5$ . No entanto, a descontinuidade provocada pela guinada acentuada fez com que alguns modelos não tivessem convergência em todas as rodadas. A TCR média de  $M_1$  é de 96,0%, enquanto para  $M_2$  é de 99,3%. Na Figura 5.7, para o BC-PLKF é observado um aumento dos indicadores de RMSE durante a manobra, com recuperação posterior, o que não é tão destacado nos cenários da Figura 5.8. Nas Figuras C.1 a C.5 é possível verificar esse fenômeno para outros modelos, sendo que alguns terminam por divergir. Desconsiderando  $I_1$  por ser essencialmente teórica,  $I_2$  proporcionou os melhores resultados para BC-PLKF, SAM-IVKF e EnKF (Tabela C.2).

Os cenários com  $M_2$  apresentaram uma certa semelhança no resultado dos modelos com o uso de  $I_2$ ,  $I_3$  e  $I_5$ . Observando a Figura 5.8, houve uma dominância de  $I_4$  no  $RMSE_r$  (que também pode ser vista nas Figuras 5.1 e 5.2), acontecendo também para o EnKF e o SAM-IVKF. Em particular para o último, as métricas obtidas foram mais discrepantes comparadas ao outros dois filtros, como pode ser observado na Figura 5.12.

Os cenários com  $M_3$  foram os que causaram maior problema para os algoritmos. Desconsiderando o MP-EKF e LP-EKF, que não tiveram estabilidade em nenhuma rodada, o EKF e AEnKF, que tiveram alguns bons resultados em outros cenários, passaram a ter mais problemas de convergência. Como já relatado na Seção 5.1, foi observado que o problema não ocorreu na segunda manobra, mas sim logo após a primeira. Com este cenário, ficou ainda mais evidente a robustez dos modelos BC-PLKF e EnKF, assim como o SAM-IVKF que se destacou em  $M_3I_4$ , onde os

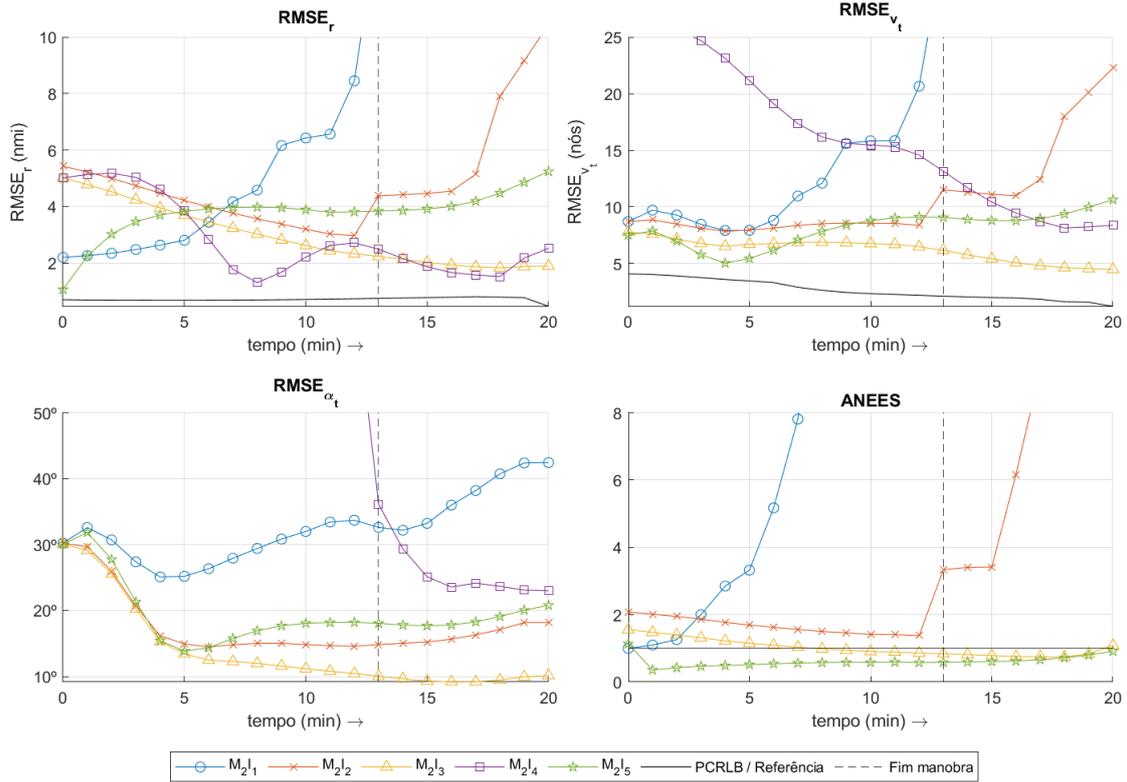


Figura 5.12: SAM-IVKF – M<sub>2</sub> – Métricas

outros dois modelos tiveram queda de desempenho por conta da inicialização. Outro ponto fundamental foi a importância do RP-EKF como forma de inicialização, que fez com que a TCR do EKF saísse de resultados inferiores a 52% para 99,9%, além de obter valores de RMSE adequados, conforme Tabela C.15 e Figura 5.13.

## 5.4 Tempo de Processamento

Os tempos de processamento de cada modelo poderia ser um fator crítico de sucesso para a implementação real destes filtros, dado que a aplicação precisaria de uma resposta com o mínimo de atraso possível. Considerando que foi trabalhado com um intervalo de amostragem de 1 minuto (como na maioria dos artigos levantados), os valores de  $\overline{\text{LPT}}$  se tornaram pouco relevantes pois não ultrapassaram 1 milissegundo no AEnKF (modelo com maior demanda de processamento) nos cenários com inicialização simples (Tabela C.4) e um máximo de 39 ms nos cenários com I<sub>5</sub> (Tabela C.5), não criando restrição para analisar diversos alvos simultâneos ou mesmo executar outros modelos ao mesmo tempo.

Apesar de não ser crítico, o  $\overline{\text{LPT}}_{\text{AEnKF}}(P^{(2)})$  foi de 0,745 ms, em média, e o  $\overline{\text{LPT}}_{\text{EnKF}}$  teve média de 0,184 ms, o que dá uma relação de 4,0 entre os dois, enquanto em [9] é dado como 1,06 esta relação.

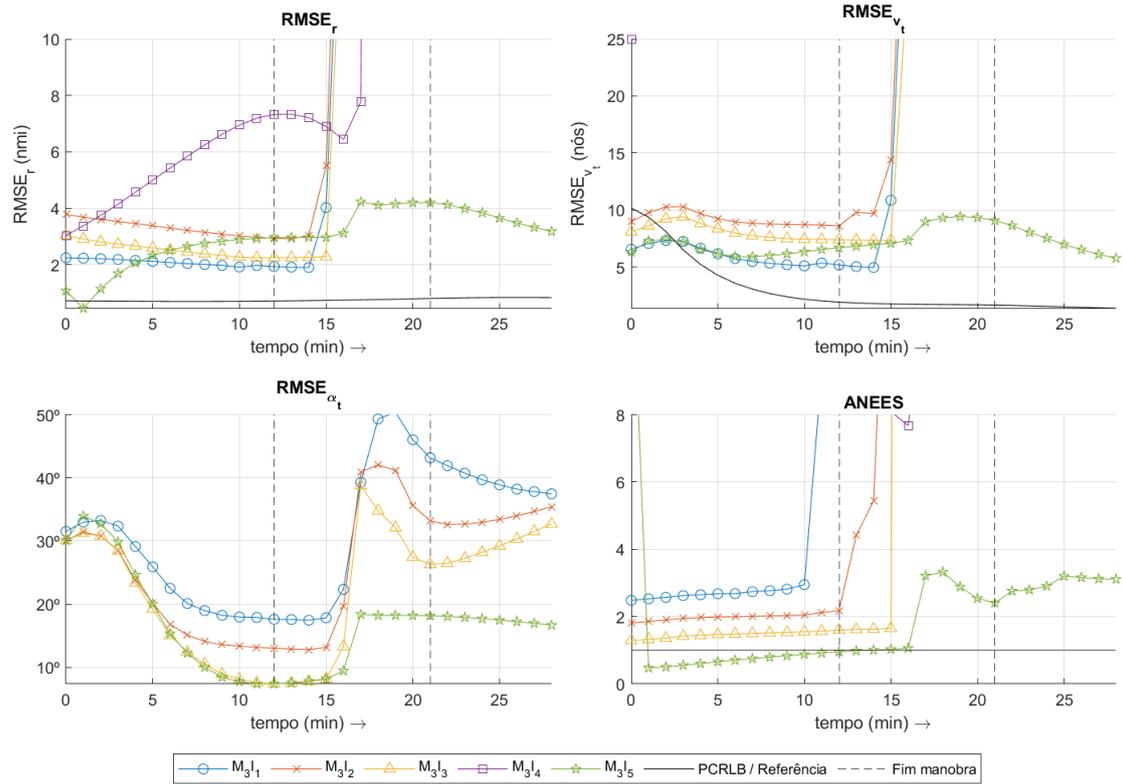


Figura 5.13: EKF –  $M_3$  – Métricas

Outro fator a ser considerado é que, em média, o uso do RP-EKF aumentou o tempo de processamento em 52,8 vezes, o que foi menos do que o esperado, já que se calculou o modelo para 64 filtros inicialmente (porém alguns são “desligados” ao longo do processo). Ainda assim, esse incremento temporal não é um fator que deva ser usado para descartar esta inicialização, que demonstrou bom desempenho, como visto na Seção 5.2. Se necessário, ainda é possível reduzir o número de intervalos para otimizar o seu processamento.

# Capítulo 6

## Conclusão

Neste trabalho, foi estudada a técnica de Filtro de Kalman para aplicação no problema de BOT. Foram avaliadas oito variações desse Filtro, sendo dois modelos aplicações diretas da teoria (EKF e EnKF) e seis modelos desenvolvidos especificamente para o problema (PLKF, BC-PLKF, SAM-IVKF, MP-EKF, LP-EKF e AEnKF). Os modelos foram aplicados em 15 cenários distintos, combinando três variações de movimentação dos navios com cinco formas de inicialização do problema. Foram realizadas simulações em MATLAB, cujos resultados foram comparados, principalmente, por meio do RMSE da distância, do RMSE da velocidade do alvo e da credibilidade do resultado medida pelo ANEES, além de avaliar a estabilidade e a convergência dos modelos.

Com base nos resultados das simulações realizadas, foi observado que os modelos levantados, em particular o BC-PLKF, o SAM-IVKF (derivado do anterior) e o EnKF podem atender a demanda de BOT para o MAGE desenvolvido pelo IPqM. O BC-PLKF se mostrou o mais consistente nos resultados e, sendo utilizado na sua versão com parametrização de distância e velocidade (RP-BC-PLKF), é possível obter boas estimativas que, embora não tenha os valores mínimos de RMSE, possuem alto índice de credibilidade.

No entanto, é possível e é desejável que o MAGE, em sua interface de apoio à tomada de decisão, forneça os resultados de outros modelos, o que inclui os já mencionados, como também algoritmos clássicos e demais filtros adaptativos. Para isso, recomenda-se o aprofundamento em outras famílias de algoritmos, como Filtro de Partículas, que nas referências encontradas apresentaram resultados equivalentes ou superiores ao Filtro de Kalman sob a restrição de maior necessidade de processamento. Dado que esta limitação não se revelou crítica até mesmo para os algoritmos de conjunto simulados, a aplicação do Filtro de Partículas pode ser viável e vantajosa.

Para a continuidade do objetivo principal desta pesquisa, os próximos passos consistem no refinamento dos modelos, aplicando-os em dados reais e em situação

de operação contínua, a fim de avaliar o comportamento dos algoritmos em relação a convergência ou mesmo estabilidade, dado que são filtros adaptativos e os resultados carregam uma “memória”. Em paralelo, deverá ser estudada a melhor forma de apresentação dos resultados sobre a fonte de sinal detectado na interface do MAGE, que envolve não apenas a parte de ergonomia cognitiva e apoio à tomada de decisão, como também a representação matemática e visual das incertezas dos dados a serem apresentados, seja por meio de elipsoides indicando regiões de maior probabilidade de localização do alvo ou outra técnica a ser levantada.

Além da avaliação de modelos alternativos, que poderia incluir um estudo de como trabalhar com resultados de algoritmos distintos de forma coordenada, trabalhos futuros poderiam envolver a evolução de algoritmos existentes, em particular o AEnKF, dado que o algoritmo precursor (EnKF) figurou entre os melhores resultados. Os demais modelos poderiam ser revisados também com o intuito de evoluir na previsão de alvos realizando manobras, não só para pequenas variações de rota como também para as demais possibilidades elencadas no trabalho de KOURKOULIS [22], de 1984, e que não foram encontradas nas referências mais atuais estudadas.

Por fim, outra possibilidade (e, de certa forma, necessidade) seria uma avaliação do impacto das manobras dos navios nos modelos, não só com um olhar sobre tipos ótimos de manobra (uma vez que essa discussão é mais relevante para submarinos que para navios de superfície), mas principalmente para identificar possíveis cenários que afetem negativamente os modelos que venham a ser implementados no MAGE, como ocorreu neste trabalho nos cenários com  $M_3$ . Desta forma, seriam estudadas formas de evitar ou mitigar esses problemas de convergência, para que em uma situação real não comprometa uma ação de defesa.

# Referências Bibliográficas

- [1] BRASIL, Ministério da Indústria, Comércio Exterior e Serviços. “Base de dados do Comex Stat. (consultas 77953 e 77956)”. 2022. Disponível em: <<http://comexstat.mdic.gov.br/pt/geral>>. Acesso em: 14 de mar. de 2023.
- [2] AGÊNCIA NACIONAL DO PETRÓLEO, GÁS NATURAL E BIO-COMBUSTÍVEIS. “Painel Dinâmico de Produção de Petróleo e Gás Natural”. Disponível em: <<https://www.gov.br/anp/pt-br/centrais-de-conteudo/paineis-dinamicos-da-anp/paineis-dinamicos-sobre-exploracao-e-producao-de-petroleo-e-gas/paineis-dinamicos-de-producao-de-petroleo-e-gas-natural>>. Acesso em: 14 de mar. de 2023.
- [3] MIGUENS, A. P. “O problema geral da navegação”. In: *Navegação: A Ciência e a Arte - volume I*, 2nd ed., cap. 1, Brasil, Diretoria de Hidrografia e Navegação - Marinha do Brasil, 2019.
- [4] DINIZ, P. S. R. “Kalman Filters”. In: *Adaptive Filtering*, 5th ed., cap. 14, Switzerland, Springer Cham, 2020. doi: 10.1007/978-3-030-29057-3.
- [5] AIDALA, V. *Behavior of the Kalman filter applied to bearings-only target motion analysis*. Relatório Técnico 4984, Naval Underwater Systems Center, Newport, R.I., USA, nov. 1976.
- [6] AIDALA, V. J. “Kalman Filter Behavior in Bearings-Only Tracking Applications”, *IEEE Transactions on Aerospace and Electronic Systems*, v. AES-15, n. 1, pp. 29–39, 1979. doi: 10.1109/TAES.1979.308793.
- [7] HOELZER, H., JOHNSON, G., COHEN, A. *Modified Polar Coordinates - The Key to Well Behaved Bearings Only Ranging*. IBM Federal System Division, Shipboard and Defense Systems, Manassas, Virginia, USA, ago. 1978.
- [8] NGUYEN, N. H., DOĞANÇAY, K. “Improved Pseudolinear Kalman Filter Algorithms for Bearings-Only Target Tracking”, *IEEE Transactions on Signal*

- Processing*, v. 65, n. 23, pp. 6119–6134, 2017. doi: 10.1109/TSP.2017.2749207.
- [9] SUN, T., XIN, M. “Bearings-Only Tracking Using Augmented Ensemble Kalman Filter”, *IEEE Transactions on Control Systems Technology*, v. 28, n. 3, pp. 1009–1016, 2020. doi: 10.1109/TCST.2018.2890370.
- [10] TIPPETT, M. K., ANDERSON, J. L., BISHOP, C. H., et al. “Ensemble Square Root Filters”, *Monthly Weather Review*, v. 131, n. 7, pp. 1485 – 1490, 2003. doi: [https://doi.org/10.1175/1520-0493\(2003\)131<1485:ESRF>2.0.CO;2](https://doi.org/10.1175/1520-0493(2003)131<1485:ESRF>2.0.CO;2). Disponível em: <[https://journals.ametsoc.org/view/journals/mwre/131/7/1520-0493\\_2003\\_131\\_1485\\_esrf\\_2.0.co\\_2.xml](https://journals.ametsoc.org/view/journals/mwre/131/7/1520-0493_2003_131_1485_esrf_2.0.co_2.xml)>.
- [11] HYDROGRAPHIC OFFICE OF U.S. NAVY. *H.O., 217: Maneuvering Board Manual*, 1st ed. Washington, 1941.
- [12] SPIESS, F. *Complete Solution of the Bearings Only Approach Problem*. Relatório Técnico 102, Marine Physical Laboratory of the Scripps Institution of Oceanography, University of California, 1953.
- [13] EKELUND, J. J. *A Means of Passive Range Determination*. Quarterly information bulletin, summer, 1958, Commander Submarine Forces, Atlantic Fleet, USA, 1958.
- [14] DOBBIE, J. M. *Complete Solution of the Bearings Only Approach Problem*. Relatório Técnico 4, Dept. of Navy, Washington D.C., USA (to Naval Analysis Programs – Office of Naval Research), 1971.
- [15] COLL, P. F. *Target motion analysis from a diesel submarine’s perspective*. Master’s thesis (operations research), Naval Postgraduate School, Monterey, CA, USA, 1994.
- [16] BAKOS, G. K. *Submarine approach and attack tactics: simulation and analysis*. Master’s thesis (operations research), Naval Postgraduate School, Monterey, CA, USA, 1995.
- [17] GENÇ, H. M. “A new solution approach for the bearing only target tracking problem”. In: *4th International Workshop on Soft Computing Applications*, pp. 95–100, 2010. doi: 10.1109/SOFA.2010.5565618.
- [18] SON, M.-J., CHO, D.-Y., CHA, J.-H., et al. “Implementation of a Tactic Manager for the Simulation of a Target Motion Analysis between a Submarine and a Surface Ship Warfare”. In: Park, J.-W., Kim, T.-G., Kim, Y.-B.

(Eds.), *AsiaSim 2007*, pp. 315–326, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN: 978-3-540-77600-0.

- [19] LIN, Y.-C., CHEN, H.-J., CHUNG, W.-H., et al. “Geometric Approach to Passive Ranging in Underwater Localization Systems”, *IEEE Access*, v. 6, pp. 54018–54032, 2018. doi: 10.1109/ACCESS.2018.2870692.
- [20] LINGREN, A. G., GONG, K. F. “Position and Velocity Estimation Via Bearing Observations”, *IEEE Transactions on Aerospace and Electronic Systems*, v. AES-14, n. 4, pp. 564–577, 1978. doi: 10.1109/TAES.1978.308681.
- [21] AIDALA, V., HAMMEL, S. “Utilization of modified polar coordinates for bearings-only tracking”, *IEEE Transactions on Automatic Control*, v. 28, n. 3, pp. 283–294, 1983. doi: 10.1109/TAC.1983.1103230.
- [22] KOURKOULIS, D. *Bearing-Only Target Tracking: maneuvering target*. Master’s thesis (operations research), Naval Postgraduate School, Monterey, CA, USA, 1984.
- [23] PEACH, N. G. *Bearings Only Tracking Using a Set of Range Parameterised Extended Kalman Filters*. Tese de Doutorado, University of Bath, Bath, UK, 1997.
- [24] BREHARD, T., CADRE, J.-P. L. “Closed-form posterior Cramer-Rao bounds for bearings-only tracking”, *IEEE Transactions on Aerospace and Electronic Systems*, v. 42, n. 4, pp. 1198–1223, 2006. doi: 10.1109/TAES.2006.314568.
- [25] LA SCALA, B., MORELANDE, M. “An analysis of the single sensor bearings-only tracking problem”. In: *2008 11th International Conference on Information Fusion*, pp. 1–6, 2008.
- [26] JAWAHAR, A., KOTESWARA RAO, S. “Modified Polar Extended Kalman Filter (MP-EKF) for Bearings - Only Target Tracking”, *Indian Journal of Science and Technology*, v. 9, pp. 1 – 5, set. 2016. doi: 10.17485/ijst/2016/v9i26/90307.
- [27] KABA, U., TEMELTAS, H. “Generalized bias compensated pseudolinear Kalman filter for colored noisy bearings-only measurements”, *Signal Processing*, v. 190, pp. 108331, 2022. ISSN: 0165-1684. doi: <https://doi.org/10.1016/j.sigpro.2021.108331>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0165168421003686>.

- [28] GUSTAFSSON, F., GUNNARSSON, F., BERGMAN, N., et al. “Particle filters for positioning, navigation, and tracking”, *IEEE Transactions on Signal Processing*, v. 50, n. 2, pp. 425–437, 2002. doi: 10.1109/78.978396.
- [29] ARULAMPALAM, M., RISTIC, B., GORDON, N., et al. “Bearings-Only Tracking of Manoeuvring Targets Using Particle Filters”, *EURASIP Journal on Applied Signal Processing*, , n. 2004:15, pp. 2351–2365, 2004. doi: 10.1155/S1110865704405095.
- [30] KÖSE, E. *Analysis of Genetic Algorithm on Bearings-Only Target Motion Analysis*. Gebze Technical University, Gebze-Kocaeli, Turkey, 2020, 5 pp.
- [31] PASSERIEUX, J., VAN CAPPEL, D. “Optimal observer maneuver for bearings-only tracking”, *IEEE Transactions on Aerospace and Electronic Systems*, v. 34, n. 3, pp. 777–788, 1998. doi: 10.1109/7.705885.
- [32] XU, Y., LIPING, L. “Single observer bearings-only tracking with the unscented Kalman filter”. In: *2004 International Conference on Communications, Circuits and Systems (IEEE Cat. No.04EX914)*, v. 2, pp. 901–905 Vol.2, 2004. doi: 10.1109/ICCCAS.2004.1346326.
- [33] CAMERON, A. C., TRIVEDI, P. K. “Linear Models”. In: *Microeconometrics: Methods and Applications*, 1 ed., cap. 4, New York, USA, Cambridge University Press, 2005.
- [34] DOĞANÇAY, K. “3D Pseudolinear Target Motion Analysis From Angle Measurements”, *IEEE Transactions on Signal Processing*, v. 63, n. 6, pp. 1570–1580, 2015. doi: 10.1109/TSP.2015.2399869.
- [35] OLADYSHKIN, S., NOWAK, W. “Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion”, *Reliability Engineering and System Safety*, v. 106, pp. 179–190, 2012. ISSN: 0951-8320. doi: <https://doi.org/10.1016/j.ress.2012.05.002>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0951832012000853>>.
- [36] FONSECA, M. M. *Arte Naval*, v. 2. 7 ed. Rio de Janeiro, Serviço de Documentação da Marinha, 2005.
- [37] TICHAVSKY, P., MURAVCHIK, C., NEHORAI, A. “Posterior Cramer-Rao bounds for discrete-time nonlinear filtering”, *IEEE Transactions on Signal Processing*, v. 46, n. 5, pp. 1386–1396, 1998. doi: 10.1109/78.668800.
- [38] LI, X. R., ZHAO, Z., LI, X.-B. “Evaluation of Estimation Algorithms: Credibility Tests”, *IEEE Transactions on Systems, Man, and Cybernetics*

- *Part A: Systems and Humans*, v. 42, n. 1, pp. 147–163, 2012. doi: 10.1109/TSMCA.2011.2158095.

[39] THE MATHWORKS INC. “MATLAB Documentation version: 9.10.0 (R2021a)”. Natick, Massachusetts, United States, 2021. Disponível em: <<https://www.mathworks.com>>. Acesso em: 01 de mai. de 2023.

# Apêndice A

## Linearização da Equação de Estados do LP-EKF

Transcrevendo as equações (3.78) e (3.79) apenas por conveniência:

$$\mathbf{z}(k) = \mathbf{f}[\mathbf{z}(k-1)] = \begin{bmatrix} f_1[\mathbf{z}(k-1)] \\ f_2[\mathbf{z}(k-1)] \\ f_3[\mathbf{z}(k-1)] \\ f_4[\mathbf{z}(k-1)] \end{bmatrix} = \begin{bmatrix} \beta(k-1) + \arctan \frac{\lambda_1(k)}{\lambda_2(k)} \\ \rho(k-1) + \frac{1}{2} \ln [\lambda_1^2(k) + \lambda_2^2(k)] \\ \frac{\lambda_3(k)\lambda_2(k) - \lambda_4(k)\lambda_1(k)}{\lambda_1^2(k) + \lambda_2^2(k)} \\ \frac{\lambda_3(k)\lambda_1(k) + \lambda_4(k)\lambda_2(k)}{\lambda_1^2(k) + \lambda_2^2(k)} \end{bmatrix} \quad (\text{A.1})$$

$$\begin{aligned} \lambda_1(k) &= \tau \dot{\beta}(k-1) - \frac{\tau}{2} e^{-\rho(k-1)} [\Delta v_{o_x}(k) \cos \beta(k-1) - \Delta v_{o_y}(k) \sin \beta(k-1)] \\ \lambda_2(k) &= 1 + \tau \dot{\rho}(k-1) - \frac{\tau}{2} e^{-\rho(k-1)} [\Delta v_{o_x}(k) \sin \beta(k-1) + \Delta v_{o_y}(k) \cos \beta(k-1)] \\ \lambda_3(k) &= \dot{\beta}(k-1) - e^{-\rho(k-1)} [\Delta v_{o_x}(k) \cos \beta(k-1) - \Delta v_{o_y}(k) \sin \beta(k-1)] \\ \lambda_4(k) &= \dot{\rho}(k-1) - e^{-\rho(k-1)} [\Delta v_{o_x}(k) \sin \beta(k-1) + \Delta v_{o_y}(k) \cos \beta(k-1)] \\ \Delta v_{o_x}(k) &= v_{o_x}(k) - v_{o_x}(k-1) \\ \Delta v_{o_y}(k) &= v_{o_y}(k) - v_{o_y}(k-1) \end{aligned} \quad (\text{A.2})$$

Calcula-se, então, as derivadas parciais de  $\lambda_1(k)$ ,  $\lambda_2(k)$ ,  $\lambda_3(k)$  e  $\lambda_4(k)$  que serão necessárias posteriormente:

$$\begin{aligned} \frac{\partial \lambda_1(k)}{\partial \beta(k-1)} &= \frac{\tau}{2} \frac{\partial \lambda_3(k)}{\partial \beta(k-1)} = \frac{\partial \lambda_2(k)}{\partial \rho(k-1)} = \frac{\tau}{2} \frac{\partial \lambda_4(k)}{\partial \rho(k-1)} = \Theta_1(k) \\ &= \frac{\tau}{2} e^{-\rho(k-1)} [\Delta v_{o_x}(k) \sin \beta(k-1) + \Delta v_{o_y}(k) \cos \beta(k-1)] \end{aligned} \quad (\text{A.3})$$

$$\begin{aligned}\frac{\partial \lambda_2(k)}{\partial \beta(k-1)} &= \frac{\tau}{2} \frac{\partial \lambda_4(k)}{\partial \beta(k-1)} = -\frac{\partial \lambda_1(k)}{\partial \rho(k-1)} = -\frac{\tau}{2} \frac{\partial \lambda_3(k)}{\partial \rho(k-1)} = \Theta_2(k) \\ &= -\frac{\tau}{2} e^{-\rho(k-1)} [\Delta v_{o_x}(k) \cos \beta(k-1) - \Delta v_{o_y}(k) \sin \beta(k-1)]\end{aligned}\quad (\text{A.4})$$

$$\frac{\partial \lambda_1(k)}{\partial \dot{\beta}(k-1)} = \frac{\partial \lambda_2(k)}{\partial \dot{\rho}(k-1)} = \tau; \quad \frac{\partial \lambda_3(k)}{\partial \dot{\beta}(k-1)} = \frac{\partial \lambda_4(k)}{\partial \dot{\rho}(k-1)} = 1 \quad (\text{A.5})$$

$$\frac{\partial \lambda_1(k)}{\partial \dot{\rho}(k-1)} = \frac{\partial \lambda_3(k)}{\partial \dot{\rho}(k-1)} = \frac{\partial \lambda_2(k)}{\partial \dot{\beta}(k-1)} = \frac{\partial \lambda_4(k)}{\partial \dot{\beta}(k-1)} = 0 \quad (\text{A.6})$$

Utilizando  $\Theta_1(k)$  e  $\Theta_2(k)$ , é possível simplificar as equações em (A.2) para facilitar a implementação computacional.

$$\begin{aligned}\lambda_1(k) &= \tau \dot{\beta}(k-1) + \Theta_2(k) \\ \lambda_2(k) &= 1 + \tau \dot{\rho}(k-1) - \Theta_1(k) \\ \lambda_3(k) &= \dot{\beta}(k-1) + \frac{2}{\tau} \Theta_2(k) \\ \lambda_4(k) &= \dot{\rho}(k-1) - \frac{2}{\tau} \Theta_1(k)\end{aligned}\quad (\text{A.7})$$

Aplicando (2.15), a matriz  $\mathbf{A}(k-1)$  é dada, então, por:

$$\begin{aligned}\mathbf{A}(k-1) &= \left. \frac{\partial \mathbf{f}[\mathbf{z}(k-1)]}{\partial \mathbf{z}(k-1)} \right|_{\mathbf{z}(k-1)=\hat{\mathbf{z}}(k-1|k-1)} \\ &= \begin{bmatrix} \frac{\partial f_1[\mathbf{z}(k-1)]}{\partial \beta(k-1)} & \frac{\partial f_1[\mathbf{z}(k-1)]}{\partial \rho(k-1)} & \frac{\partial f_1[\mathbf{z}(k-1)]}{\partial \dot{\beta}(k-1)} & \frac{\partial f_1[\mathbf{z}(k-1)]}{\partial \dot{\rho}(k-1)} \\ \frac{\partial f_2[\mathbf{z}(k-1)]}{\partial \beta(k-1)} & \frac{\partial f_2[\mathbf{z}(k-1)]}{\partial \rho(k-1)} & \frac{\partial f_2[\mathbf{z}(k-1)]}{\partial \dot{\beta}(k-1)} & \frac{\partial f_2[\mathbf{z}(k-1)]}{\partial \dot{\rho}(k-1)} \\ \frac{\partial f_3[\mathbf{z}(k-1)]}{\partial \beta(k-1)} & \frac{\partial f_3[\mathbf{z}(k-1)]}{\partial \rho(k-1)} & \frac{\partial f_3[\mathbf{z}(k-1)]}{\partial \dot{\beta}(k-1)} & \frac{\partial f_3[\mathbf{z}(k-1)]}{\partial \dot{\rho}(k-1)} \\ \frac{\partial f_4[\mathbf{z}(k-1)]}{\partial \beta(k-1)} & \frac{\partial f_4[\mathbf{z}(k-1)]}{\partial \rho(k-1)} & \frac{\partial f_4[\mathbf{z}(k-1)]}{\partial \dot{\beta}(k-1)} & \frac{\partial f_4[\mathbf{z}(k-1)]}{\partial \dot{\rho}(k-1)} \end{bmatrix}_{\mathbf{z}(k-1)=\hat{\mathbf{z}}(k-1|k-1)}\end{aligned}\quad (\text{A.8})$$

Para o cálculo das derivadas parciais, a partir deste ponto serão omitidos os índices temporais  $(k)$  e  $(k-1)$  para não sobrecarregar as equações:

$$\frac{\partial f_1}{\partial \beta} = 1 + \frac{\Theta_1 \lambda_2 - \lambda_1 \Theta_2}{\lambda_1^2 + \lambda_2^2} \quad (\text{A.9})$$

$$\frac{\partial f_1}{\partial \rho} = \frac{-\Theta_2 \lambda_2 - \lambda_1 \Theta_1}{\lambda_1^2 + \lambda_2^2} \quad (\text{A.10})$$

$$\frac{\partial f_1}{\partial \dot{\beta}} = \frac{\tau \lambda_2 - \lambda_1 \cdot 0}{\lambda_1^2 + \lambda_2^2} = \frac{\tau \lambda_2}{\lambda_1^2 + \lambda_2^2} \quad (\text{A.11})$$

$$\frac{\partial f_1}{\partial \dot{\rho}} = \frac{0 \cdot \lambda_2 - \lambda_1 \tau}{\lambda_1^2 + \lambda_2^2} = \frac{-\lambda_1 \tau}{\lambda_1^2 + \lambda_2^2} \quad (\text{A.12})$$

$$\frac{\partial f_2}{\partial \dot{\beta}} = \frac{\lambda_1 \Theta_1 + \lambda_2 \Theta_2}{\lambda_1^2 + \lambda_2^2} = -\frac{\partial f_1}{\partial \rho} \quad (\text{A.13})$$

$$\frac{\partial f_2}{\partial \rho} = 1 + \frac{-\lambda_1 \Theta_2 + \lambda_2 \Theta_1}{\lambda_1^2 + \lambda_2^2} = \frac{\partial f_1}{\partial \beta} \quad (\text{A.14})$$

$$\frac{\partial f_2}{\partial \dot{\beta}} = \frac{\lambda_1 \tau + \lambda_2 \cdot 0}{\lambda_1^2 + \lambda_2^2} = -\frac{\partial f_1}{\partial \dot{\rho}} \quad (\text{A.15})$$

$$\frac{\partial f_2}{\partial \dot{\rho}} = \frac{\lambda_1 \cdot 0 + \lambda_2 \tau}{\lambda_1^2 + \lambda_2^2} = \frac{\partial f_1}{\partial \dot{\beta}} \quad (\text{A.16})$$

$$\frac{\partial f_3}{\partial \dot{\beta}} = \frac{(\frac{2}{\tau} \Theta_1 \lambda_2 + \lambda_3 \Theta_2 - \frac{2}{\tau} \Theta_2 \lambda_1 - \lambda_4 \Theta_1) (\lambda_1^2 + \lambda_2^2) - (\lambda_3 \lambda_2 - \lambda_4 \lambda_1) (2\lambda_1 \Theta_1 + 2\lambda_2 \Theta_2)}{(\lambda_1^2 + \lambda_2^2)^2} \quad (\text{A.17})$$

$$\frac{\partial f_3}{\partial \rho} = \frac{(-\frac{2}{\tau} \Theta_2 \lambda_2 + \lambda_3 \Theta_1 - \frac{2}{\tau} \Theta_1 \lambda_1 + \lambda_4 \Theta_2) (\lambda_1^2 + \lambda_2^2) - (\lambda_3 \lambda_2 - \lambda_4 \lambda_1) (-2\lambda_1 \Theta_2 + 2\lambda_2 \Theta_1)}{(\lambda_1^2 + \lambda_2^2)^2} \quad (\text{A.18})$$

$$\begin{aligned} \frac{\partial f_3}{\partial \dot{\beta}} &= \frac{(1 \cdot \lambda_2 + \lambda_3 \cdot 0 - 0 \cdot \lambda_1 - \lambda_4 \tau) (\lambda_1^2 + \lambda_2^2) - (\lambda_3 \lambda_2 - \lambda_4 \lambda_1) (2\lambda_1 \tau + 2\lambda_2 \cdot 0)}{(\lambda_1^2 + \lambda_2^2)^2} \\ &= \frac{(\lambda_2 - \lambda_4 \tau) (\lambda_1^2 + \lambda_2^2) - 2\lambda_1 \tau (\lambda_3 \lambda_2 - \lambda_4 \lambda_1)}{(\lambda_1^2 + \lambda_2^2)^2} \end{aligned} \quad (\text{A.19})$$

$$\begin{aligned} \frac{\partial f_3}{\partial \dot{\rho}} &= \frac{(0 \cdot \lambda_2 + \lambda_3 \tau - 1 \cdot \lambda_1 - \lambda_4 \cdot 0) (\lambda_1^2 + \lambda_2^2) - (\lambda_3 \lambda_2 - \lambda_4 \lambda_1) (2\lambda_1 \cdot 0 + 2\lambda_2 \tau)}{(\lambda_1^2 + \lambda_2^2)^2} \\ &= \frac{(\lambda_3 \tau - \lambda_1) (\lambda_1^2 + \lambda_2^2) - 2\lambda_2 \tau (\lambda_3 \lambda_2 - \lambda_4 \lambda_1)}{(\lambda_1^2 + \lambda_2^2)^2} \end{aligned} \quad (\text{A.20})$$

$$\frac{\partial f_4}{\partial \dot{\beta}} = \frac{(\frac{2}{\tau} \Theta_1 \lambda_1 + \lambda_3 \Theta_1 + \frac{2}{\tau} \Theta_2 \lambda_2 + \lambda_4 \Theta_2) (\lambda_1^2 + \lambda_2^2) - (\lambda_3 \lambda_1 + \lambda_4 \lambda_2) (2\lambda_1 \Theta_1 + 2\lambda_2 \Theta_2)}{(\lambda_1^2 + \lambda_2^2)^2} \quad (\text{A.21})$$

$$\frac{\partial f_4}{\partial \rho} = \frac{(-\frac{2}{\tau} \Theta_2 \lambda_1 - \lambda_3 \Theta_2 + \frac{2}{\tau} \Theta_1 \lambda_2 + \lambda_4 \Theta_1) (\lambda_1^2 + \lambda_2^2) - (\lambda_3 \lambda_1 + \lambda_4 \lambda_2) (-2\lambda_1 \Theta_2 + 2\lambda_2 \Theta_1)}{(\lambda_1^2 + \lambda_2^2)^2} \quad (\text{A.22})$$

$$\begin{aligned} \frac{\partial f_4}{\partial \dot{\beta}} &= \frac{(1 \cdot \lambda_1 + \lambda_3 \tau + 0 \cdot \lambda_2 + \lambda_4 \cdot 0) (\lambda_1^2 + \lambda_2^2) - (\lambda_3 \lambda_1 + \lambda_4 \lambda_2) (2\lambda_1 \tau + 2\lambda_2 \cdot 0)}{(\lambda_1^2 + \lambda_2^2)^2} \\ &= \frac{(\lambda_1 + \lambda_3 \tau) (\lambda_1^2 + \lambda_2^2) - 2\lambda_1 \tau (\lambda_3 \lambda_1 + \lambda_4 \lambda_2)}{(\lambda_1^2 + \lambda_2^2)^2} \end{aligned} \quad (\text{A.23})$$

$$\begin{aligned} \frac{\partial f_4}{\partial \dot{\rho}} &= \frac{(0 \cdot \lambda_1 + \lambda_3 \cdot 0 + 1 \cdot \lambda_2 + \lambda_4 \tau) (\lambda_1^2 + \lambda_2^2) - (\lambda_3 \lambda_1 + \lambda_4 \lambda_2) (2\lambda_1 \cdot 0 + 2\lambda_2 \tau)}{(\lambda_1^2 + \lambda_2^2)^2} \\ &= \frac{(\lambda_2 + \lambda_4 \tau) (\lambda_1^2 + \lambda_2^2) - 2\lambda_2 \tau (\lambda_3 \lambda_1 + \lambda_4 \lambda_2)}{(\lambda_1^2 + \lambda_2^2)^2} \end{aligned} \quad (\text{A.24})$$

# Apêndice B

## Códigos em MATLAB

Os códigos de *loop* dos modelos implementados em MATLAB como funções encontram-se neste Apêndice, onde cada seção corresponde a um arquivo, cujo nome deve ser o mesmo do nome da seção. Estas funções precisam ser chamadas por um código principal. Este código e os demais utilizados podem ser obtidos por *download* pelo *link*: [https://drive.google.com/file/d/1ygoLr9cwhpMBQzInbAxeZa8d7zWSMWM\\_/view?usp=share\\_link](https://drive.google.com/file/d/1ygoLr9cwhpMBQzInbAxeZa8d7zWSMWM_/view?usp=share_link).

### B.1 EKF\_loop.m

```
function [x_k_k, Pe_k_k, tempo] = EKF_loop(A, x_km1_km1, Pe_km1_km1, ...
                                         beta_k, Delta_v_o_k, sigma2_k, tau)

% Loop do algoritmo de EKF aplicado a BOT
% ===== ENTRADAS =====
% A          matriz de transição de estado
% x_km1_km1  estimativa da variável de estado x no instante
%            anterior: x_hat(k-1|k-1)
% Pe_km1_km1 matriz de covariância do erro de x no instante anterior:
%            Pe(k-1|k-1)
% beta_k     medição da marcação no instante atual (em radianos):
%            beta_tilde(k)
% Delta_v_o_k vetor variação de velocidade do observador no instante
%            atual em relação ao anterior (em nós): v_o(k) - v_o(k-1)
% sigma2_k   variância do erro de medição eta(k) no instante atual
%            (em rad^2): sigma^2(k)
% tau        taxa de amostragem (em horas)
% ===== SAÍDAS =====
% x_k_k      estimativa da variável de estado x no instante atual:
%            x_hat(k|k)
% Pe_k_k     matriz de covariância do erro de x no instante atual:
%            Pe(k|k)
% tempo      tempo de processamento (em segundos)
```

```

tic

w_k = [(tau/2) * Delta_v_o_k; Delta_v_o_k];
x_k_km1 = A * x_km1_km1 - w_k;
Pe_k_km1 = A * Pe_km1_km1 * A';
aux_den = x_k_km1(1)^2 + x_k_km1(2)^2;
c_k = [x_k_km1(2)/aux_den; -x_k_km1(1)/aux_den; 0; 0];
g_k = Pe_k_km1 * c_k / (c_k' * Pe_k_km1 * c_k + sigma2_k);
x_k_k = x_k_km1 + g_k * (beta_k - arctan_BOT(x_k_km1(1:2,:),1,1));
Pe_k_k = (eye(4) - g_k * c_k') * Pe_k_km1;

tempo = toc;

end

```

## B.2 PLKF\_loop.m

```

function [x_k_k, Pe_k_k, tempo] = PLKF_loop(A, x_km1_km1, Pe_km1_km1, ...
                                         beta_k, Delta_v_o_k, sigma2_k, tau)

% Loop do algoritmo PLKF aplicado a BOT
% ===== ENTRADAS =====
% A          matriz de transição de estado
% x_km1_km1  estimativa da variável de estado x no instante
%            anterior: x_hat(k-1|k-1)
% Pe_km1_km1 matriz de covariância do erro de x no instante anterior:
%            Pe(k-1|k-1)
% beta_k     medição da marcação no instante atual (em radianos):
%            beta_tilde(k)
% Delta_v_o_k vetor variação de velocidade do observador no instante
%            atual em relação ao anterior (em nós): v_o(k) - v_o(k-1)
% sigma2_k   variância do erro de medição eta(k) no instante atual
%            (em rad^2): sigma^2(k)
% tau        taxa de amostragem (em horas)
% ===== SAÍDAS =====
% x_k_k      estimativa da variável de estado x no instante atual:
%            x_hat(k|k)
% Pe_k_k     matriz de covariância do erro de x no instante atual:
%            Pe(k|k)
% tempo      tempo de processamento (em segundos)

tic

w_k = [(tau/2) * Delta_v_o_k; Delta_v_o_k];
x_k_km1 = A * x_km1_km1 - w_k;
Pe_k_km1 = A * Pe_km1_km1 * A';
c_k = [cos(beta_k); -sin(beta_k); 0; 0];
g_k = Pe_k_km1 * c_k / (c_k' * Pe_k_km1 * c_k + sigma2_k);
x_k_k = x_k_km1 - g_k * c_k' * x_k_km1;
Pe_k_k = (eye(4) - g_k * c_k') * Pe_k_km1;

```

```

    tempo = toc;

end

```

## B.3 BCPLKF\_loop.m

```

function [x_k_k, Pe_k_k, tempo] = BCPLKF_loop(A, x_km1_km1, Pe_km1_km1, ...
        beta_k, Delta_v_o_k, sigma2_k, tau, Q)

% Loop do algoritmo BC-PLKF aplicado a BOT
% ===== ENTRADAS =====
% A            matriz de transição de estado
% x_km1_km1    estimativa da variável de estado x no instante
%              anterior: x_hat(k-1|k-1)
% Pe_km1_km1   matriz de covariância do erro de x no instante anterior:
%              Pe(k-1|k-1)
% beta_k       medição da marcação no instante atual (em radianos):
%              beta_tilde(k)
% Delta_v_o_k  vetor variação de velocidade do observador no instante
%              atual em relação ao anterior (em nós): v_o(k) - v_o(k-1)
% sigma2_k     variância do erro de medição eta(k) no instante atual
%              (em rad^2): sigma^2(k)
% tau          taxa de amostragem (em horas)
% Q            matriz de covariância da perturbação que provoca
%              mudança de rota do alvo
% ===== SAÍDAS =====
% x_k_k        estimativa da variável de estado x no instante atual:
%              x_hat(k|k)
% Pe_k_k       matriz de covariância do erro de x no instante atual:
%              Pe(k|k)
% tempo        tempo de processamento (em segundos)

tic

w_k = [(tau/2) * Delta_v_o_k; Delta_v_o_k];
x_k_km1 = A * x_km1_km1 - w_k;
Pe_k_km1 = A * Pe_km1_km1 * A' + Q;
c_k = [cos(beta_k); -sin(beta_k); 0; 0];
Pnu_k = (x_k_km1(1,1)^2 + x_k_km1(2,1)^2) * sigma2_k;
g_k = Pe_k_km1 * c_k / (c_k' * Pe_k_km1 * c_k + Pnu_k);
x_k_k = x_k_km1 - g_k * c_k' * x_k_km1;
Pe_k_k = (eye(4) - g_k * c_k') * Pe_k_km1;
x_k_k = x_k_k + Pe_k_k / Pnu_k * sigma2_k * [x_k_k(1:2,1); 0; 0];

tempo = toc;

end

```

## B.4 SAMIVKF\_loop.m

```

function [x_k_k, Pe_k_k, tempo] = SAMIVKF_loop(A, x_km1_km1, Pe_km1_km1, ...
        beta_k, Delta_v_o_k, sigma2_k, tau, Q, kappa)

% Loop do algoritmo SAM-IVKF aplicado a BOT
% ===== ENTRADAS =====
% A          matriz de transição de estado
% x_km1_km1  estimativa da variável de estado x no instante
%            anterior: x_hat(k-1|k-1)
% Pe_km1_km1 matriz de covariância do erro de x no instante anterior:
%            Pe(k-1|k-1)
% beta_k     medição da marcação no instante atual (em radianos):
%            beta_tilde(k)
% Delta_v_o_k vetor variação de velocidade do observador no instante
%            atual em relação ao anterior (em nós): v_o(k) - v_o(k-1)
% sigma2_k   variância do erro de medição eta(k) no instante atual
%            (em rad^2): sigma^2(k)
% tau        taxa de amostragem (em horas)
% Q          matriz de covariância da perturbação que provoca
%            mudança de rota do alvo
% kappa      fator múltiplo de sigma_n no interv. [2,4] para SAM
% ===== SAÍDAS =====
% x_k_k      estimativa da variável de estado x no instante atual:
%            x_hat(k|k)
% Pe_k_k     matriz de covariância do erro de x no instante atual:
%            Pe(k|k)
% tempo      tempo de processamento (em segundos)

tic

w_k = [(tau/2) * Delta_v_o_k; Delta_v_o_k];
x_k_km1 = A * x_km1_km1 - w_k;
Pe_k_km1 = A * Pe_km1_km1 * A' + Q;
c_k = [cos(beta_k); -sin(beta_k); 0; 0];
Pnu_k = (x_k_km1(1,1)^2 + x_k_km1(2,1)^2) * sigma2_k;
g_k = Pe_k_km1 * c_k / (c_k' * Pe_k_km1 * c_k + Pnu_k);
x_k_k = x_k_km1 - g_k * c_k' * x_k_km1;
Pe_k_k = (eye(4) - g_k * c_k') * Pe_k_km1;
x_k_k = x_k_k + Pe_k_k / Pnu_k * sigma2_k * [x_k_k(1:2,1); 0; 0];
beta_BC = arctan_BOT(x_k_k(1:2,1), 1, 1);
if abs(DiferencaAngulo(beta_BC - beta_k, 1)) < kappa * sqrt(sigma2_k)
    c_IV = [cos(beta_BC); -sin(beta_BC); 0; 0];
    g_IV_k = Pe_k_km1 * c_IV / (c_k' * Pe_k_km1 * c_IV + Pnu_k);
    x_k_k = x_k_km1 - g_IV_k * c_k' * x_k_km1;
    Pe_k_k = (eye(4) - g_IV_k * c_k') * Pe_k_km1;
end

tempo = toc;

end

```

## B.5 EnKF\_loop.m

```
function [x_k_k, Pe_k_k, tempo] = EnKF_loop(A, x_km1_km1, Pe_km1_km1, ...
    beta_k, Delta_v_o_k, sigma2_k, tau, Q, J)
% Loop do algoritmo de EnKF aplicado a BOT
% ===== ENTRADAS =====
% A          matriz de transição de estado
% x_km1_km1  estimativa da variável de estado x no instante
%            anterior: x_hat(k-1|k-1)
% Pe_km1_km1 matriz de covariância do erro de x no instante anterior:
%            Pe(k-1|k-1)
% beta_k     medição da marcação no instante atual (em radianos):
%            beta_tilde(k)
% Delta_v_o_k vetor variação de velocidade do observador no instante
%            atual em relação ao anterior (em nós): v_o(k) - v_o(k-1)
% sigma2_k   variância do erro de medição eta(k) no instante atual
%            (em rad^2): sigma^2(k)
% tau        taxa de amostragem (em horas)
% Q          matriz de covariância do processo aleatório para geração
%            do conjunto de variáveis de estado
% J          número de elementos do conjunto
% ===== SAÍDAS =====
% x_k_k      estimativa da variável de estado x no instante atual:
%            x_hat(k|k)
% Pe_k_k     matriz de covariância do erro de x no instante atual:
%            Pe(k|k)
% tempo      tempo de processamento (em segundos)

tic

w_k = [(tau/2) * Delta_v_o_k; Delta_v_o_k];
x_k_km1 = A * x_km1_km1 - w_k;
x_k_km1_j = repmat(x_k_km1,1,J) + [mvnrnd(zeros(1, 4), Q, J)]';
x_k_km1 = mean(x_k_km1_j, 2);
Pe_k_km1 = A * Pe_km1_km1 * A' + cov(x_k_km1_j');
beta_j = repmat(beta_k,1,J) + randn(1, J) .* sqrt(sigma2_k);
beta_bar = mean(beta_j);
P_beta = var(beta_j);
aux_den = x_k_km1(1)^2 + x_k_km1(2)^2;
c_k = [x_k_km1(2)/aux_den; -x_k_km1(1)/aux_den; 0; 0];
g_k = Pe_k_km1 * c_k / (c_k' * Pe_k_km1 * c_k + P_beta);
x_k_k = x_k_km1 + g_k * (beta_bar - arctan_BOT(x_k_km1(1:2,:),1,1));
Pe_k_k = (eye(4) - g_k * c_k') * Pe_k_km1;
tempo = toc;

end
```

## B.6 AEnKF\_loop.m

```
function [x_k_k, x_k_k_j, Pe_k_k, tempo] = AEnKF_loop(A, x_km1_km1_j, ...
    Pe_km1_km1, beta_k, Delta_v_o_k, sigma2_k, tau, J, grauP)
% Loop do algoritmo de AEnKF aplicado a BOT para 2o e 3o grau
% ===== ENTRADAS =====
% A          matriz de transição de estado
% x_km1_km1_j conjunto de estimativas da variável de estado x no
%             instante anterior: x_hat_j(k-1|k-1)
% Pe_km1_km1 matriz de covariância do erro de x no instante anterior:
%             Pe(k-1|k-1)
% beta_k     medição da marcação no instante atual (em radianos):
%             beta_tilde(k)
% Delta_v_o_k vetor variação de velocidade do observador no instante
%             atual em relação ao anterior (em nós): v_o(k) - v_o(k-1)
% sigma2_k   variância do erro de medição eta(k) no instante atual
%             (em rad^2): sigma^2(k)
% tau        taxa de amostragem (em horas)
% Q          matriz de covariância do processo aleatório para geração
%             do conjunto de variáveis de estado
% J          número de elementos do conjunto
% grauP      grau do maior polinômio (apenas 2 ou 3 programados)
% ===== SAÍDAS =====
% x_k_k      estimativas da variável de estado x (média do conjunto
%             no instante atual: x_bar(k|k)
% x_k_k_j    conjunto de estimativas da variável de estado x no
%             instante atual: x_hat_j(k|k)
% Pe_k_k     matriz de covariância do erro de x no instante atual:
%             Pe(k|k)
% tempo      tempo de processamento (em segundos)

tic

w_k = repmat([(tau/2) * Delta_v_o_k; Delta_v_o_k],1,J);
x_k_km1_j = A * x_km1_km1_j - w_k + [mvnrnd(zeros(1, 4), Q, J)]';
x_k_km1 = mean(x_k_km1_j, 2);
Pe_k_km1 = cov(x_k_km1_j', 1) + A * Pe_km1_km1 * A';
beta_j = arctan_BOT(x_k_km1_j(1:2,:),1,1) + randn(1, J) .* sqrt(sigma2_k);
beta_bar = mean(beta_j);
mu_2 = var(beta_j, 1);
beta_j_norm = (beta_j - repmat(beta_bar,1,J)) ./ sqrt(mu_2);
beta_k_norm = (beta_k - beta_bar) / sqrt(mu_2);
m_i = mean(beta_j_norm.^3);
p_l_2 = [-1, -m_i(1), 1];
auxBetaPol = [ones(1, J); beta_j_norm; beta_j_norm.^2];
P_2_j = p_l_2 * auxBetaPol;
phi_j = [P_2_j];
P_2 = p_l_2 * [1; beta_k_norm; beta_k_norm^2];
```

```

phi = [P_2];
if grauP >= 3
    grauP = 3;
    m_i = [m_i; mean([beta_j_norm.^4; beta_j_norm.^5], 2)];
    p_l_3 = [m_i(1)^2 - m_i(1)^3 + m_i(1) * m_i(2) - m_i(3), ...
            -m_i(1) * m_i(3) + m_i(1)^2 - m_i(2) + m_i(1) * m_i(2), ...
            -m_i(1) * m_i(2) + m_i(3) - m_i(1), 1 - m_i(1) + m_i(1)^2];
    auxBetaPol = [auxBetaPol; beta_j_norm.^3];
    P_3_j = p_l_3 * auxBetaPol;
    phi_j = [phi_j; P_3_j];
    P_3 = p_l_3 * [1; beta_k_norm; beta_k_norm^2; beta_k_norm^3];
    phi = [phi; P_3];
end
yaug_hat_j = [beta_j; phi_j];
yaug_hat_bar = [beta_bar; mean(phi_j, 2)];
yaug = [beta_k; phi];
Py = cov(yaug_hat_j');
Pxy = zeros(4, grauP);
for j=1:J
    Pxy = Pxy + (x_k_km1_j(:,j)-x_k_km1)*(yaug_hat_j(:,j)-yaug_hat_bar)';
end
Pxy = Pxy ./ (J-1);
G_k = Pxy / Py;
x_k_k = x_k_km1 + G_k * (yaug - yaug_hat_bar);
x_k_k_j = x_k_km1_j + G_k * (repmat(yaug,1,J) - yaug_hat_j);
Pe_k_k = Pe_k_km1 - Pxy / Py * Pxy';
tempo = toc;
end

```

## B.7 MPEKF\_loop.m

```

function [z_k_k, Pe_k_k, tempo] = MPEKF_loop(z_km1_km1, Pe_km1_km1, ...
                                             beta_k, Delta_v_o_k, sigma2_k, tau)
% Loop do algoritmo de MP-EKF aplicado a BOT
% ===== ENTRADAS =====
% z_km1_km1      estimativa da variável de estado z no instante
%                anterior: z_hat(k-1|k-1)
% Pe_km1_km1     matriz de covariância do erro de x no instante anterior:
%                Pe_MP(k-1|k-1)
% beta_k         medição da marcação no instante atual (em radianos):
%                beta_tilde(k)
% Delta_v_o_k    vetor variação de velocidade tangencial e radial do
%                observador no instante atual em relação ao anterior (em nós):
%                [v_o_tan(k); vo_rad(k)] - [v_o_tan(k-1); vo_rad(k-1)]
% sigma2_k       variância do erro de medição eta(k) no instante atual
%                (em rad^2): sigma^2(k)

```

```

% tau          taxa de amostragem (em horas)
% ===== SAÍDAS =====
% z_k_k        estimativa da variável de estado z no instante atual:
%              z_hat(k|k)
% Pe_k_k       matriz de covariância do erro de z no instante atual:
%              Pe_MP(k|k)
% tempo        tempo de processamento (em segundos)

tic

A_delta = eye(4);
A_delta(2:3, 4) = - Delta_v_o_k(:, 1);
z_km1_km1_m = A_delta * z_km1_km1;

% variáveis auxiliares
dotBetaTau = z_km1_km1_m(2,1) * tau;
dotRhoTauM1 = 1 + z_km1_km1_m(3,1) * tau;
ellTau = z_km1_km1_m(4,1) * tau;
Denominador = dotRhoTauM1^2 + dotBetaTau^2;

z_k_km1 = zeros(4,1);
z_k_km1(1,1) = z_km1_km1_m(1,1) + atan(dotBetaTau / dotRhoTauM1);
z_k_km1(2,1) = z_km1_km1_m(2,1) / Denominador;
z_k_km1(3,1) = (z_km1_km1_m(3,1) + (z_km1_km1_m(3,1)^2 + z_km1_km1_m(2,1)^2)
...
    * tau) / Denominador;
z_k_km1(4,1) = z_km1_km1_m(4,1) / Denominador^(1/2);
A_MRU = eye(4);
A_MRU(1,2) = dotRhoTauM1 * tau / Denominador;
A_MRU(1,3) = - dotBetaTau * tau / Denominador;
A_MRU(2,2) = (dotRhoTauM1^2 - dotBetaTau^2) / Denominador^2;
A_MRU(2,3) = - 2 * dotBetaTau * dotRhoTauM1 / Denominador^2;
A_MRU(3,2) = - A_MRU(2,3);
A_MRU(3,3) = A_MRU(2,2);
A_MRU(4,2) = - dotBetaTau * ellTau / Denominador^1.5;
A_MRU(4,3) = - dotRhoTauM1 * ellTau / Denominador^1.5;
A_MRU(4,4) = Denominador^-0.5;
Pe_k_km1 = A_MRU * A_delta * Pe_km1_km1 * A_delta' * A_MRU';
c_k = [1; 0; 0; 0];
g_k = Pe_k_km1 * c_k / (c_k' * Pe_k_km1 * c_k + sigma2_k);
z_k_k = z_k_km1 + g_k * (beta_k - z_k_km1(1,1));
Pe_k_k = (eye(4) - g_k * c_k') * Pe_k_km1;
tempo = toc;

end

```

## B.8 LPEKF\_loop.m

```
function [z_k_k, Pe_k_k, tempo] = LPEKF_loop(z_km1_km1, Pe_km1_km1, ...
```

```

                                beta_k, Delta_v_o_k, sigma2_k, tau)
% Loop do algoritmo de MP-EKF aplicado a BOT
% ===== ENTRADAS =====
% z_km1_km1      estimativa da variável de estado z no instante
%                anterior: z_hat(k-1|k-1)
% Pe_km1_km1     matriz de covariância do erro de x no instante anterior:
%                Pe_MP(k-1|k-1)
% beta_k         medição da marcação no instante atual (em radianos):
%                beta_tilde(k)
% Delta_v_o_k    vetor variação de velocidade do observador no instante
%                atual em relação ao anterior (em nós): v_o(k) - v_o(k-1)
% sigma2_k       variância do erro de medição eta(k) no instante atual
%                (em rad^2): sigma^2(k)
% tau            taxa de amostragem (em horas)
% ===== SAÍDAS =====
% z_k_k          estimativa da variável de estado z no instante atual:
%                z_hat(k|k)
% Pe_k_k         matriz de covariância do erro de z no instante atual:
%                Pe_MP(k|k)
% tempo          tempo de processamento (em segundos)

tic

% variáveis auxiliares
Theta1 = tau/2*exp(-z_km1_km1(2,1))*(Delta_v_o_k(1,1)*sin(z_km1_km1(1,1))...
    + Delta_v_o_k(2,1)*cos(z_km1_km1(1,1)));
Theta2 = -tau/2*exp(-z_km1_km1(2,1))*(Delta_v_o_k(1,1)*cos(z_km1_km1(1,1))...
    - Delta_v_o_k(2,1)*sin(z_km1_km1(1,1)));
lambda1 = tau * z_km1_km1(3,1) + Theta2;
lambda2 = 1 + tau * z_km1_km1(4,1) - Theta1;
lambda3 = z_km1_km1(3,1) + 2/tau*Theta2;
lambda4 = z_km1_km1(4,1) - 2/tau*Theta1;
somaQuadL1L2 = lambda1^2 + lambda2^2;

z_k_km1 = zeros(4,1);
z_k_km1(1,1) = z_km1_km1(1,1) + atan(lambda1/lambda2);
z_k_km1(2,1) = z_km1_km1(2,1) + 0.5 * log(somaQuadL1L2);
z_k_km1(3,1) = (lambda3*lambda2 - lambda4*lambda1)/somaQuadL1L2;
z_k_km1(4,1) = (lambda3*lambda1 + lambda4*lambda2)/somaQuadL1L2;

A_k = zeros(4);
A_k(1,1) = 1 + (Theta1*lambda2 - lambda1*Theta2)/somaQuadL1L2;
A_k(1,2) = (-Theta2*lambda2 - lambda1*Theta1)/somaQuadL1L2;
A_k(1,3) = (tau*lambda2)/somaQuadL1L2;
A_k(1,4) = (-tau*lambda1)/somaQuadL1L2;

A_k(2,1) = -A_k(1,2);
A_k(2,2) = A_k(1,1);
A_k(2,3) = -A_k(1,4);
A_k(2,4) = A_k(1,3);

```

```

A_k(3,1) = ((2/tau*Theta1*lambda2 + lambda3*Theta2 - 2/tau*Theta2*lambda1...
- lambda4*Theta1)*somaQuadL1L2 - (lambda3*lambda2 - lambda4*lambda1)...
*(2*lambda1*Theta1 + 2*lambda2*Theta2))/somaQuadL1L2^2;
A_k(3,2) = ((-2/tau*Theta2*lambda2 + lambda3*Theta1 - 2/tau*Theta1*lambda1...
+ lambda4*Theta2)*somaQuadL1L2 - (lambda3*lambda2 - lambda4*lambda1)...
*(-2*lambda1*Theta2 + 2*lambda2*Theta1))/somaQuadL1L2^2;
A_k(3,3) = ((lambda2 - lambda4*tau)*somaQuadL1L2 - 2*lambda1*tau*...
(lambda3*lambda2 - lambda4*lambda1))/somaQuadL1L2^2;
A_k(3,4) = ((lambda3*tau - lambda1)*somaQuadL1L2 - 2*lambda2*tau*...
(lambda3*lambda2 - lambda4*lambda1))/somaQuadL1L2^2;

A_k(4,1) = ((2/tau*Theta1*lambda1 + lambda3*Theta1 + 2/tau*Theta2*lambda2...
+ lambda4*Theta2)*somaQuadL1L2 - (lambda3*lambda1 + lambda4*lambda2)...
*(2*lambda1*Theta1 + 2*lambda2*Theta2))/somaQuadL1L2^2;
A_k(4,2) = ((-2/tau*Theta2*lambda1 - lambda3*Theta2 + 2/tau*Theta1*lambda2...
+ lambda4*Theta1)*somaQuadL1L2 - (lambda3*lambda1 + lambda4*lambda2)...
*(-2*lambda1*Theta2 + 2*lambda2*Theta1))/somaQuadL1L2^2;
A_k(4,3) = ((lambda1 + lambda3*tau)*somaQuadL1L2 - 2*lambda1*tau*...
(lambda3*lambda1 + lambda4*lambda2))/somaQuadL1L2^2;
A_k(4,4) = ((lambda2 + lambda4*tau)*somaQuadL1L2 - 2*lambda2*tau*...
(lambda3*lambda1 + lambda4*lambda2))/somaQuadL1L2^2;

Pe_k_km1 = A_k * Pe_km1_km1 * A_k';
c_k = [1; 0; 0; 0];
g_k = Pe_k_km1 * c_k / (c_k' * Pe_k_km1 * c_k + sigma2_k);
z_k_k = z_k_km1 + g_k * (beta_k - z_k_km1(1,1));
Pe_k_k = (eye(4) - g_k * c_k') * Pe_k_km1;

tempo = toc;

```

end

## B.9 RP\_EKF\_loop.m

```

function [x_k_k, x_k_k_i, Pe_k_k, Pe_k_k_i, tempo, xi_k] = RP_EKF_loop(A,...
x_km1_km1, Pe_km1_km1, beta_k, Delta_v_o_k, sigma2_k, tau, xi_km1, limite)
% Loop do algoritmo de RP-EKF aplicado a BOT
% ===== ENTRADAS =====
% A          matriz de transição de estado
% x_km1_km1  estimativa da variável de estado x no instante
%            anterior: x_hat_i(k-1|k-1)
% Pe_km1_km1 matriz de covariância do erro de x no instante anterior:
%            Pe_i(k-1|k-1)
% beta_k     medição da marcação no instante atual (em radianos):
%            beta_tilde(k)
% Delta_v_o_k vetor variação de velocidade do observador no instante
%            atual em relação ao anterior (em nós): v_o(k) - v_o(k-1)
% sigma2_k   variância do erro de medição eta(k) no instante atual
%            (em rad^2): sigma^2(k)

```

```

% tau          taxa de amostragem (em horas)
% xi_km1       peso de cada modelo i do instante anterior
% limite       valor mínimo de peso para considerar o modelo
% ===== SAÍDAS =====
% x_k_k, x_k_k_i   estimativa da variável de estado x no instante atual
%                  médio e para cada filtro i: x_hat(k|k) e x_hat_i(k|k)
% Pe_k_k        matriz de covariância do erro de x no instante atual:
%                  para cada filtro i: Pe_i(k|k)
% tempo         tempo de processamento (em segundos)
% xi_k         peso de cada modelo i atualizado

t_inic = tic;
N_i = length(xi_km1);
x_k_k_i = zeros(4,N_i);
Pe_k_k_i = zeros(4,4,N_i);

for i = 1:N_i
    if xi_km1(i) >= limite
        [x_k_k_i(:,i),Pe_k_k_i(:, :,i), ] = EKF_loop(A, x_km1_km1(:,i),...
            Pe_km1_km1(:, :,i), beta_k, Delta_v_o_k, sigma2_k, tau);
    end
end

xi_k = AtualizaPesosRP(xi_km1, x_k_k_i, Pe_k_k_i, beta_k, sigma2_k, ...
    limite, 0);
x_k_k = x_k_k_i * xi_k';
Pe_k_k = zeros(4);
for i = 1:N_i
    if xi_k(i) >= limite
        Pe_k_k = Pe_k_k + xi_k(i) .* (Pe_k_k_i(:, :,i) + ...
            (x_k_k_i(:,i) - x_k_k)*(x_k_k_i(:,i) - x_k_k)');
    end
end

tempo = toc(t_inic);

end

```

## B.10 RP\_PLKF\_loop.m

```

function [x_k_k, x_k_k_i, Pe_k_k, Pe_k_k_i, tempo, xi_k] = RP_PLKF_loop(A,...
    x_km1_km1, Pe_km1_km1, beta_k, Delta_v_o_k, sigma2_k, tau, xi_km1, limite)
% Loop do algoritmo de RP-PLKF aplicado a BOT
% ===== ENTRADAS =====
% A          matriz de transição de estado
% x_km1_km1  estimativa da variável de estado x no instante
%             anterior: x_hat_i(k-1|k-1)
% Pe_km1_km1 matriz de covariância do erro de x no instante anterior:
%             Pe_i(k-1|k-1)
% beta_k     medição da marcação no instante atual (em radianos):

```

```

%          beta_tilde(k)
% Delta_v_o_k  vetor variação de velocidade do observador no instante
%              atual em relação ao anterior (em nós):  v_o(k) - v_o(k-1)
% sigma2_k     variância do erro de medição eta(k) no instante atual
%              (em rad^2):  sigma^2(k)
% tau          taxa de amostragem (em horas)
% xi_km1       peso de cada modelo i do instante anterior
% limite       valor mínimo de peso para considerar o modelo
% ===== SAÍDAS =====
% x_k_k, x_k_k_i  estimativa da variável de estado x no instante atual
%                 médio e para cada filtro i:  x_hat(k|k) e x_hat_i(k|k)
% Pe_k_k        matriz de covariância do erro de x no instante atual:
%                 para cada filtro i:  Pe_i(k|k)
% tempo         tempo de processamento (em segundos)
% xi_k          peso de cada modelo i atualizado

t_inic = tic;
N_i = length(xi_km1);
x_k_k_i = zeros(4,N_i);
Pe_k_k_i = zeros(4,4,N_i);

for i = 1:N_i
    if xi_km1(i) >= limite
        [x_k_k_i(:,i),Pe_k_k_i(:, :,i), ] = PLKF_loop(A, x_km1_km1(:,i),...
            Pe_km1_km1(:, :,i), beta_k, Delta_v_o_k, sigma2_k, tau);
    end
end
xi_k = AtualizaPesosRP(xi_km1, x_k_k_i, Pe_k_k_i, beta_k, sigma2_k, ...
    limite, 0);
x_k_k = x_k_k_i * xi_k';
Pe_k_k = zeros(4);
for i = 1:N_i
    if xi_k(i) >= limite
        Pe_k_k = Pe_k_k + xi_k(i) .* (Pe_k_k_i(:, :,i) + ...
            (x_k_k_i(:,i) - x_k_k)*(x_k_k_i(:,i) - x_k_k)');
    end
end
tempo = toc(t_inic);

end

```

## B.11 RP\_BCPLKF\_loop.m

```

function [x_k_k, x_k_k_i, Pe_k_k, Pe_k_k_i, tempo, xi_k] = RP_BCPLKF_loop(A,...
    x_km1_km1, Pe_km1_km1, beta_k, Delta_v_o_k, sigma2_k, tau, Q, xi_km1, limite)
% Loop do algoritmo de RP-BC-PLKF aplicado a BOT
% ===== ENTRADAS =====
% A          matriz de transição de estado

```

```

% x_km1_km1      estimativa da variável de estado x no instante
%                anterior: x_hat_i(k-1|k-1)
% Pe_km1_km1     matriz de covariância do erro de x no instante anterior:
%                Pe_i(k-1|k-1)
% beta_k         medição da marcação no instante atual (em radianos):
%                beta_tilde(k)
% Delta_v_o_k    vetor variação de velocidade do observador no instante
%                atual em relação ao anterior (em nós): v_o(k) - v_o(k-1)
% sigma2_k       variância do erro de medição eta(k) no instante atual
%                (em rad^2): sigma^2(k)
% tau            taxa de amostragem (em horas)
% Q              matriz de covariância da perturbação que provoca
%                mudança de rota do alvo
% xi_km1         peso de cada modelo i do instante anterior
% limite         valor mínimo de peso para considerar o modelo
% ===== SAÍDAS =====
% x_k_k, x_k_k_i  estimativa da variável de estado x no instante atual
%                médio e para cada filtro i: x_hat(k|k) e x_hat_i(k|k)
% Pe_k_k         matriz de covariância do erro de x no instante atual:
%                para cada filtro i: Pe_i(k|k)
% tempo          tempo de processamento (em segundos)
% xi_k           peso de cada modelo i atualizado

t_inic = tic;
N_i = length(xi_km1);
x_k_k_i = zeros(4,N_i);
Pe_k_k_i = zeros(4,4,N_i);

for i = 1:N_i
    if xi_km1(i) >= limite
        [x_k_k_i(:,i),Pe_k_k_i(:, :,i), ] = BCPLKF_loop(A, x_km1_km1(:,i),...
            Pe_km1_km1(:, :,i), beta_k, Delta_v_o_k, sigma2_k, tau, Q);
    end
end
xi_k = AtualizaPesosRP(xi_km1, x_k_k_i, Pe_k_k_i, beta_k, sigma2_k, ...
    limite, 0);
x_k_k = x_k_k_i * xi_k';
Pe_k_k = zeros(4);
for i = 1:N_i
    if xi_k(i) >= limite
        Pe_k_k = Pe_k_k + xi_k(i) .* (Pe_k_k_i(:, :,i) + ...
            (x_k_k_i(:,i) - x_k_k)*(x_k_k_i(:,i) - x_k_k)');
    end
end
tempo = toc(t_inic);
end

```

## B.12 RP\_SAMIVKF\_loop.m

```

function [x_k_k, x_k_k_i, Pe_k_k, Pe_k_k_i, tempo, xi_k] = RP_SAMIVKF_loop(A,...
    x_km1_km1, Pe_km1_km1, beta_k, Delta_v_o_k, sigma2_k, tau, Q, kappa, ...
    xi_km1, limite)
% Loop do algoritmo de RP-SAM-IVKF aplicado a BOT
% ===== ENTRADAS =====
% A          matriz de transição de estado
% x_km1_km1  estimativa da variável de estado x no instante
%            anterior: x_hat_i(k-1|k-1)
% Pe_km1_km1 matriz de covariância do erro de x no instante anterior:
%            Pe_i(k-1|k-1)
% beta_k     medição da marcação no instante atual (em radianos):
%            beta_tilde(k)
% Delta_v_o_k vetor variação de velocidade do observador no instante
%            atual em relação ao anterior (em nós): v_o(k) - v_o(k-1)
% sigma2_k   variância do erro de medição eta(k) no instante atual
%            (em rad^2): sigma^2(k)
% tau        taxa de amostragem (em horas)
% Q          matriz de covariância da perturbação que provoca
%            mudança de rota do alvo
% kappa      fator múltiplo de sigma_n no interv. [2,4] para SAM
% xi_km1     peso de cada modelo i do instante anterior
% limite     valor mínimo de peso para considerar o modelo
% ===== SAÍDAS =====
% x_k_k, x_k_k_i  estimativa da variável de estado x no instante atual
%                médio e para cada filtro i: x_hat(k|k) e x_hat_i(k|k)
% Pe_k_k         matriz de covariância do erro de x no instante atual:
%                para cada filtro i: Pe_i(k|k)
% tempo         tempo de processamento (em segundos)
% xi_k          peso de cada modelo i atualizado

t_inic = tic;
N_i = length(xi_km1);
x_k_k_i = zeros(4,N_i);
Pe_k_k_i = zeros(4,4,N_i);

for i = 1:N_i
    if xi_km1(i) >= limite
        [x_k_k_i(:,i),Pe_k_k_i(:, :,i), ] = SAMIVKF_loop(A, x_km1_km1(:,i),...
            Pe_km1_km1(:, :,i), beta_k, Delta_v_o_k, sigma2_k, tau, Q, kappa);
    end
end

xi_k = AtualizaPesosRP(xi_km1, x_k_k_i, Pe_k_k_i, beta_k, sigma2_k, ...
    limite, 0);
x_k_k = x_k_k_i * xi_k';
Pe_k_k = zeros(4);
for i = 1:N_i

```

```

        if xi_k(i) >= limite
            Pe_k_k = Pe_k_k + xi_k(i) .* (Pe_k_k_i(:, :, i) + ...
                (x_k_k_i(:, i) - x_k_k)*(x_k_k_i(:, i) - x_k_k)');
        end
    end
    end
    tempo = toc(t_inic);
end

```

## B.13 RP\_EnKF\_loop.m

```

function [x_k_k, x_k_k_i, Pe_k_k, Pe_k_k_i, tempo, xi_k] = RP_EnKF_loop(A,...
    x_km1_km1, Pe_km1_km1, beta_k, Delta_v_o_k, sigma2_k, tau, Q, J, xi_km1,...
    limite)
% Loop do algoritmo de RP-EnKF aplicado a BOT
% ===== ENTRADAS =====
% A            matriz de transição de estado
% x_km1_km1    estimativa da variável de estado x no instante
%              anterior: x_hat_i(k-1|k-1)
% Pe_km1_km1   matriz de covariância do erro de x no instante anterior:
%              Pe_i(k-1|k-1)
% beta_k       medição da marcação no instante atual (em radianos):
%              beta_tilde(k)
% Delta_v_o_k  vetor variação de velocidade do observador no instante
%              atual em relação ao anterior (em nós): v_o(k) - v_o(k-1)
% sigma2_k     variância do erro de medição eta(k) no instante atual
%              (em rad^2): sigma^2(k)
% tau          taxa de amostragem (em horas)
% Q            matriz de covariância do processo aleatório para geração
%              do conjunto de variáveis de estado
% J            número de elementos do conjunto
% xi_km1       peso de cada modelo i do instante anterior
% limite       valor mínimo de peso para considerar o modelo
% ===== SAÍDAS =====
% x_k_k, x_k_k_i  estimativa da variável de estado x no instante atual
%                 médio e para cada filtro i: x_hat(k|k) e x_hat_i(k|k)
% Pe_k_k        matriz de covariância do erro de x no instante atual:
%                 para cada filtro i: Pe_i(k|k)
% tempo         tempo de processamento (em segundos)
% xi_k          peso de cada modelo i atualizado

t_inic = tic;
N_i = length(xi_km1);
x_k_k_i = zeros(4, N_i);
Pe_k_k_i = zeros(4, 4, N_i);

for i = 1:N_i
    if xi_km1(i) >= limite

```

```

        [x_k_k_i(:,i),Pe_k_k_i(:,i), ] = EnKF_loop(A, x_km1_km1(:,i),...
            Pe_km1_km1(:,i), beta_k, Delta_v_o_k, sigma2_k, tau, Q, J);
    end
end
xi_k = AtualizaPesosRP(xi_km1, x_k_k_i, Pe_k_k_i, beta_k, sigma2_k, ...
    limite, 0);
x_k_k = x_k_k_i * xi_k';
Pe_k_k = zeros(4);
for i = 1:N_i
    if xi_k(i) >= limite
        Pe_k_k = Pe_k_k + xi_k(i) .* (Pe_k_k_i(:,i) + ...
            (x_k_k_i(:,i) - x_k_k)*(x_k_k_i(:,i) - x_k_k)');
    end
end
end
tempo = toc(t_inic);
end

```

## B.14 RP\_AEnKF\_loop.m

```

function [x_k_k, x_k_k_j_i, Pe_k_k, Pe_k_k_i, tempo, xi_k] = RP_AEnKF_loop(A,...
    x_km1_km1_j, Pe_km1_km1, beta_k, Delta_v_o_k, sigma2_k, tau, Q, J, grauP,...
    xi_km1, limite)
% Loop do algoritmo de RP-EnKF aplicado a BOT
% ===== ENTRADAS =====
% A            matriz de transição de estado
% x_km1_km1_j  conjunto de estimativas da variável de estado x no
%              instante anterior: x_hat_j,i(k-1|k-1)
% Pe_km1_km1   matriz de covariância do erro de x no instante anterior:
%              Pe_i(k-1|k-1)
% beta_k       medição da marcação no instante atual (em radianos):
%              beta_tilde(k)
% Delta_v_o_k  vetor variação de velocidade do observador no instante
%              atual em relação ao anterior(em nós): v_o(k) - v_o(k-1)
% sigma2_k     variância do erro de medição eta(k) no instante atual
%              (em rad^2): sigma^2(k)
% tau         taxa de amostragem (em horas)
% Q           matriz de covariância do processo aleatório para geração
%              do conjunto de variáveis de estado
% J           número de elementos do conjunto
% grauP       grau do maior polinômio (apenas 2 ou 3 programados)
% xi_km1      peso de cada modelo i do instante anterior
% limite      valor mínimo de peso para considerar o modelo
% ===== SAÍDAS =====
% x_k_k, x_k_k_j_i  estimativa da variável de estado x no instante atual
%                  médio e para cada filtro i: x_hat(k|k) e x_hat_j,i(k|k)
% Pe_k_k         matriz de covariância do erro de x no instante atual:

```

```

%           para cada filtro i: Pe_i(k|k)
% tempo           tempo de processamento (em segundos)
% xi_k           peso de cada modelo i atualizado

t_inic = tic;
N_i = length(xi_km1);
x_k_k_i = zeros(4,N_i);
x_k_k_j_i = zeros(4,J,N_i);
Pe_k_k_i = zeros(4,4,N_i);

for i = 1:N_i
    if xi_km1(i) >= limite
        [x_k_k_i(:,i), x_k_k_j_i(:, :, i), Pe_k_k_i(:, :, i), ] = AEnKF_loop...
            (A, x_km1_km1_j(:, :, i), Pe_km1_km1(:, :, i), beta_k, ...
            Delta_v_o_k, sigma2_k, tau, Q, J, grauP);
    end
end
xi_k = AtualizaPesosRP(xi_km1, x_k_k_i, Pe_k_k_i, beta_k, sigma2_k, ...
    limite, 0);

x_k_k = x_k_k_i * xi_k';
Pe_k_k = zeros(4);
for i = 1:N_i
    if xi_k(i) >= limite
        Pe_k_k = Pe_k_k + xi_k(i) .* (Pe_k_k_i(:, :, i) + ...
            (x_k_k_i(:, i) - x_k_k)*(x_k_k_i(:, i) - x_k_k)');
    end
end
tempo = toc(t_inic);

end

```

## B.15 RP\_MPEKF\_loop.m

```

function [z_k_k, z_k_k_i, Pe_k_k, Pe_k_k_i, tempo, xi_k] = RP_MPEKF_loop(...
    z_km1_km1, Pe_km1_km1, beta_k, Delta_v_o_k, sigma2_k, tau, xi_km1, limite)
% Loop do algoritmo de RP-MP-EKF aplicado a BOT
% ===== ENTRADAS =====
% A           matriz de transição de estado
% z_km1_km1   estimativa da variável de estado z no instante
%             anterior: z_hat_i(k-1|k-1)
% Pe_km1_km1  matriz de covariância do erro de z no instante anterior:
%             Pe_i(k-1|k-1)
% beta_k      medição da marcação no instante atual (em radianos):
%             beta_tilde(k)
% Delta_v_o_k vetor variação de velocidade tangencial e radial do
%             observador no instante atual em relação ao anterior (em nós):
%             [v_o_tan(k); vo_rad(k)] - [v_o_tan(k-1); vo_rad(k-1)]
% sigma2_k    variância do erro de medição eta(k) no instante atual

```

```

%           (em rad^2):  sigma^2(k)
% tau       taxa de amostragem (em horas)
% xi_km1    peso de cada modelo i do instante anterior
% limite    valor mínimo de peso para considerar o modelo
% ===== SAÍDAS =====
% z_k_k, z_k_k_i  estimativa da variável de estado z no instante atual
%                 médio e para cada filtro i:  z_hat(k|k) e z_hat_i(k|k)
% Pe_k_k      matriz de covariância do erro de z no instante atual:
%             para cada filtro i:  Pe_i(k|k)
% tempo       tempo de processamento (em segundos)
% xi_k       peso de cada modelo i atualizado

t_inic = tic;
N_i = length(xi_km1);
z_k_k_i = zeros(4,N_i);
Pe_k_k_i = zeros(4,4,N_i);
for i = 1:N_i
    if xi_km1(i) >= limite
        [z_k_k_i(:,i),Pe_k_k_i(:, :,i), ] = MPEKF_loop(z_km1_km1(:,i),...
            Pe_km1_km1(:, :,i), beta_k, Delta_v_o_k, sigma2_k, tau);
    end
end
xi_k = AtualizaPesosRP(xi_km1, z_k_k_i, Pe_k_k_i, beta_k, sigma2_k, ...
    limite, 1);
z_k_k = z_k_k_i * xi_k';
Pe_k_k = zeros(4);
for i = 1:N_i
    if xi_k(i) >= limite
        Pe_k_k = Pe_k_k + xi_k(i) .* (Pe_k_k_i(:, :,i) + ...
            (z_k_k_i(:,i) - z_k_k)*(z_k_k_i(:,i) - z_k_k)');
    end
end
tempo = toc(t_inic);
end

```

## B.16 RP\_LPEKF\_loop.m

```

function [z_k_k, z_k_k_i, Pe_k_k, Pe_k_k_i, tempo, xi_k] = RP_LPEKF_loop(...
    z_km1_km1, Pe_km1_km1, beta_k, Delta_v_o_k, sigma2_k, tau, xi_km1, limite)
% Loop do algoritmo de RP-LP-EKF aplicado a BOT
% ===== ENTRADAS =====
% A           matriz de transição de estado
% z_km1_km1   estimativa da variável de estado z no instante
%             anterior:  z_hat_i(k-1|k-1)
% Pe_km1_km1  matriz de covariância do erro de z no instante anterior:
%             Pe_i(k-1|k-1)
% beta_k      medição da marcação no instante atual (em radianos):

```

```

%          beta_tilde(k)
% Delta_v_o_k  vetor variação de velocidade do observador no instante
%              atual em relação ao anterior (em nós):  v_o(k) - v_o(k-1)
% sigma2_k     variância do erro de medição eta(k) no instante atual
%              (em rad^2):  sigma^2(k)
% tau          taxa de amostragem (em horas)
% xi_km1       peso de cada modelo i do instante anterior
% limite       valor mínimo de peso para considerar o modelo
% ===== SAÍDAS =====
% z_k_k, z_k_k_i  estimativa da variável de estado z no instante atual
%                 médio e para cada filtro i:  z_hat(k|k) e z_hat_i(k|k)
% Pe_k_k       matriz de covariância do erro de z no instante atual:
%              para cada filtro i:  Pe_i(k|k)
% tempo        tempo de processamento (em segundos)
% xi_k         peso de cada modelo i atualizado

t_inic = tic;
N_i = length(xi_km1);
z_k_k_i = zeros(4,N_i);
Pe_k_k_i = zeros(4,4,N_i);
for i = 1:N_i
    if xi_km1(i) >= limite
        [z_k_k_i(:,i),Pe_k_k_i(:, :,i), ] = LPEKF_loop(z_km1_km1(:,i),...
            Pe_km1_km1(:, :,i), beta_k, Delta_v_o_k, sigma2_k, tau);
    end
end
xi_k = AtualizaPesosRP(xi_km1, z_k_k_i, Pe_k_k_i, beta_k, sigma2_k, ...
    limite, 1);
z_k_k = z_k_k_i * xi_k';
Pe_k_k = zeros(4);
for i = 1:N_i
    if xi_k(i) >= limite
        Pe_k_k = Pe_k_k + xi_k(i) .* (Pe_k_k_i(:, :,i) + ...
            (z_k_k_i(:,i) - z_k_k)*(z_k_k_i(:,i) - z_k_k)');
    end
end
tempo = toc(t_inic);
end

```

## B.17 AtualizaPesosRP.m

```

function [xi_i_novo] = AtualizaPesosRP(xi_i, x_k_k_i, Pe_k_k_i, beta_k, ...
    sigma_n2, limite, polar)
% Atualiza o peso dos filtros para o Range Parameterized EKF e variações
% ===== ENTRADAS =====
% xi_i         Pesos anteriores dos filtros
% x_k_k_i      Variáveis x_i(k|k) (ou z_i(k|k) para MP ou LP-EKF)

```

```

% Pe_k_k_i      Matriz de cov do erro P_e_i(k|k)
% beta_k        Medição atual (em rad)
% sigma_n2      Variância do erro de medição
% limite        Peso mínimo para considerar o modelo
% polar         1 para variável em coordenada polar (MP ou LP)
% ===== SAÍDA =====
% sigma_n2      Pesos atualizados

N_i = length(xi_i);
xi_i_novo = zeros(size(xi_i));

for i = 1:N_i
    if xi_i(i) >= limite
        if polar == 1
            beta_hat = x_k_k_i(1,i);
            sigma_beta2 = Pe_k_k_i(1,1,i);
        else
            beta_hat = arctan_BOT(x_k_k_i(1:2,i),1,1);
            sigma_beta2 = (x_k_k_i(2,i)^2*abs(Pe_k_k_i(1,1,i)) + ...
                x_k_k_i(1,i)^2*abs(Pe_k_k_i(2,2,i))) / (x_k_k_i(2,i)^2 + ...
                x_k_k_i(1,i)^2)^2;
        end
        xi_i_novo(i) = xi_i(i) / sqrt(2*pi*(sigma_n2+sigma_beta2)) * exp...
            (-0.5*DiferencaAngulo(beta_k-beta_hat,1)^2/(sigma_n2+...
            sigma_beta2));
    else
        xi_i_novo(i) = 0;
    end
end
end
if sum(xi_i_novo)==0; xi_i_novo = xi_i; end
xi_i_novo = xi_i_novo ./ sum(xi_i_novo);
end

```

## B.18 arctan\_BOT.m

```

function angulo = arctan_BOT(vetor, radiano, centro0)
% Calcula o arco tangente para 4 quadrantes para um vetor ou
% matriz de vetores para aplicação em marcação ou rumo
% ===== ENTRADAS =====
% vetor          valor ou array de vetores posição relativa (r_x, r_y) ou
%                velocidade (v_x, v_y)
% radiano        1 se saída em radianos, 0 para graus
% centro0        1 se resultado em [-pi, pi[, 0 para [0, 2pi[
% ===== SAÍDA =====
% angulo         marcação (para posição) ou rumo (para velocidade),
%                na unidade escolhida

if ndims(vetor)==3

```

```
        angulo = atan2(vetor(1, :, :), vetor(2, :, :));
    else
        angulo = atan2(vetor(1, :), vetor(2, :));
    end
    if centro0 == 0
        angulo = ((angulo < 0) .* (angulo + 2 * pi)) + ((angulo >= 0) .* (angulo));
    end
    if radiano = 1
        angulo = angulo ./ pi * 180;
    end
end
```

# Apêndice C

## Resultados Detalhados por Cenário

### C.1 Movimentação 1, Inicialização 1

Tabela C.1: Métricas –  $M_1I_1$  (médias temporais)

Modelo	$\overline{RMSE}_r$	$\overline{RMSE}_{v_t}$	$\overline{RMSE}_{\alpha_t}$	$\overline{ANEES}$	$\overline{TPT}$	$\overline{LPT}$	$\overline{RPT}$	SR	TCR
EKF	4,582	21,685	29,08°	$1,44 \cdot 10^3$	0,159	0,010	1,000	100%	99,95%
PLKF	<b>2,731</b>	<b>10,774</b>	30,56°	<b>54,936</b>	0,097	0,006	0,612	100%	100%
BC-PLKF	<b>1,714</b>	<b>5,003</b>	<b>15,91°</b>	<b>11,123</b>	0,237	0,015	1,495	100%	100%
SAM-IVKF	21,702	40,915	<b>21,47°</b>	<b>49,407</b>	0,298	0,019	1,880	100%	99,10%
EnKF	<b>1,549</b>	<b>7,248</b>	29,51°	343,190	2,704	0,169	17,041	100%	100%
AEnKF ( $P^{(2)}$ )	8,826	55,990	<b>28,58°</b>	540,988	11,182	0,699	70,465	100%	97,00%
MP-EKF	230,812	$1,48 \cdot 10^3$	111,37°	$7,17 \cdot 10^6$	0,259	0,016	1,635	100%	95,95%
LP-EKF	$4,88 \cdot 10^3$	$2,01 \cdot 10^5$	112,54°	$1,64 \cdot 10^7$	0,103	0,006	0,648	100%	97,35%
PCRLB	0,312	2,434							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados **em negrito**

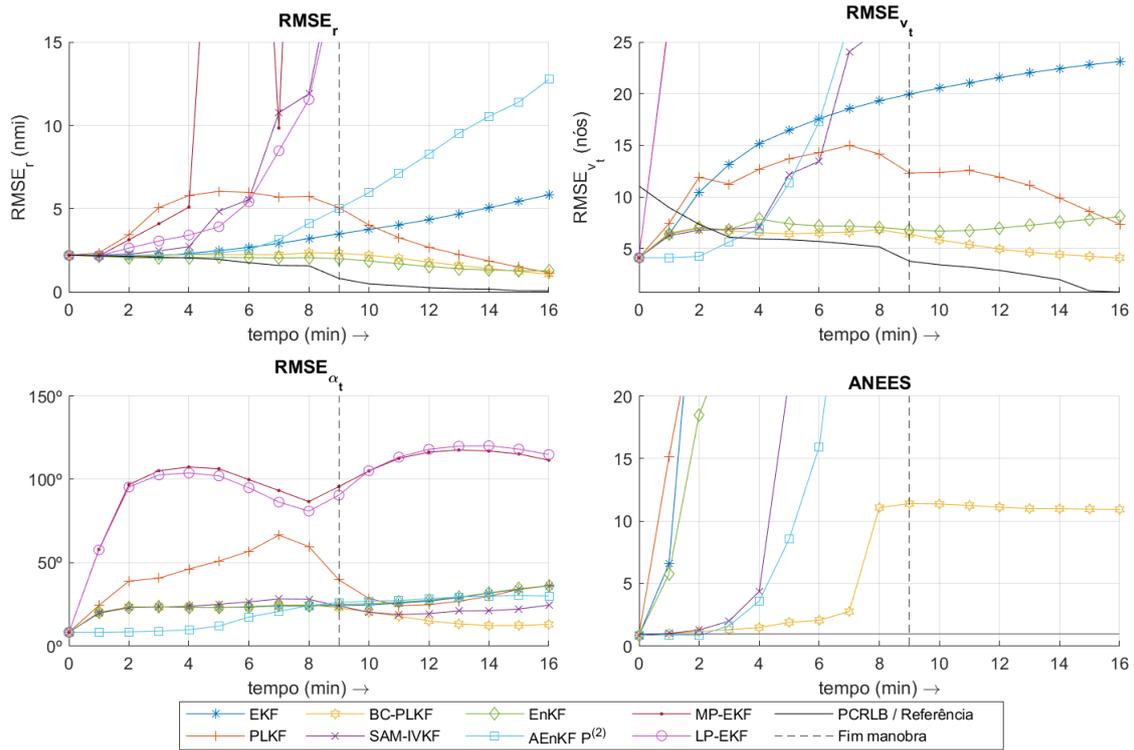


Figura C.1: Métricas –  $M_1I_1$

## C.2 Movimentação 1, Inicialização 2

Tabela C.2: Métricas –  $M_1I_2$  (médias temporais)

Modelo	$\overline{RMSE}_r$	$\overline{RMSE}_{v_t}$	$\overline{RMSE}_{\alpha_t}$	$\overline{ANEES}$	$\overline{TPT}$	$\overline{LPT}$	$\overline{RPT}$	SR	TCR
EKF	4,573	<b>7,892</b>	19,73°	3,372	0,175	0,011	1,000	100%	100%
PLKF	<b>2,844</b>	11,787	25,90°	54,478	0,107	0,007	0,609	100%	100%
BC-PLKF	<b>4,306</b>	8,208	<b>19,33°</b>	<b>2,905</b>	0,259	0,016	1,478	100%	100%
SAM-IVKF	<b>3,140</b>	<b>7,620</b>	<b>14,85°</b>	<b>1,995</b>	0,331	0,021	1,887	100%	100%
EnKF	4,576	7,892	19,75°	3,380	3,011	0,188	17,159	100%	100%
AEnKF ( $P^{(2)}$ )	6,728	<b>5,182</b>	<b>4,42°</b>	<b>2,816</b>	12,155	0,760	69,261	100%	100%
MP-EKF	$1,42 \cdot 10^3$	$6,45 \cdot 10^3$	110,66°	$3,98 \cdot 10^8$	0,285	0,018	1,623	100%	52,05%
LP-EKF	$3,36 \cdot 10^5$	$5,73 \cdot 10^6$	122,41°	$9,85 \cdot 10^8$	0,114	0,007	0,651	100%	77,15%
PCRLB	0,313	2,487							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados **em negrito**

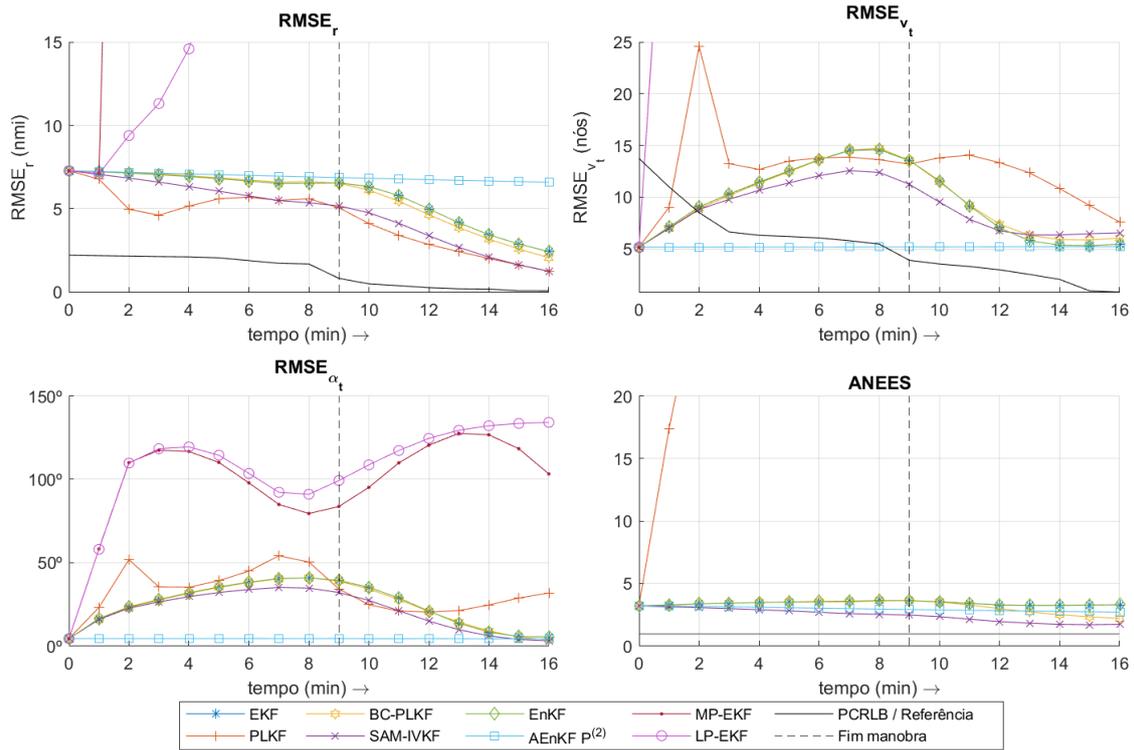


Figura C.2: Métricas –  $M_1I_2$

### C.3 Movimentação 1, Inicialização 3

Tabela C.3: Métricas –  $M_1I_3$  (médias temporais)

Modelo	$\overline{RMSE}_r$	$\overline{RMSE}_{v_t}$	$\overline{RMSE}_{\alpha_t}$	$\overline{ANEES}$	$\overline{TPT}$	$\overline{LPT}$	$\overline{RPT}$	SR	TCR
EKF	4,224	<b>6,495</b>	18,12°	2,909	0,205	0,013	1,000	100%	100%
PLKF	<b>2,840</b>	11,733	25,60°	54,408	0,126	0,008	0,617	100%	100%
BC-PLKF	<b>4,011</b>	6,986	<b>17,79°</b>	<b>2,471</b>	0,300	0,019	1,467	100%	100%
SAM-IVKF	<b>2,545</b>	<b>6,158</b>	<b>12,60°</b>	<b>1,415</b>	0,391	0,024	1,910	100%	100%
EnKF	4,229	6,501	18,16°	2,919	3,522	0,220	17,208	100%	100%
AEnKF ( $P^{(2)}$ )	6,378	<b>3,088</b>	<b>2,44°</b>	<b>2,312</b>	13,911	0,869	67,975	100%	100%
MP-EKF	$1,16 \cdot 10^3$	$4,88 \cdot 10^3$	112,04°	$1,17 \cdot 10^8$	0,331	0,021	1,617	100%	48,50%
LP-EKF	$2,58 \cdot 10^6$	$2,33 \cdot 10^7$	124,11°	$9,27 \cdot 10^8$	0,136	0,008	0,662	100%	76,75%
PCRLB	0,242	1,860							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados **em negrito**

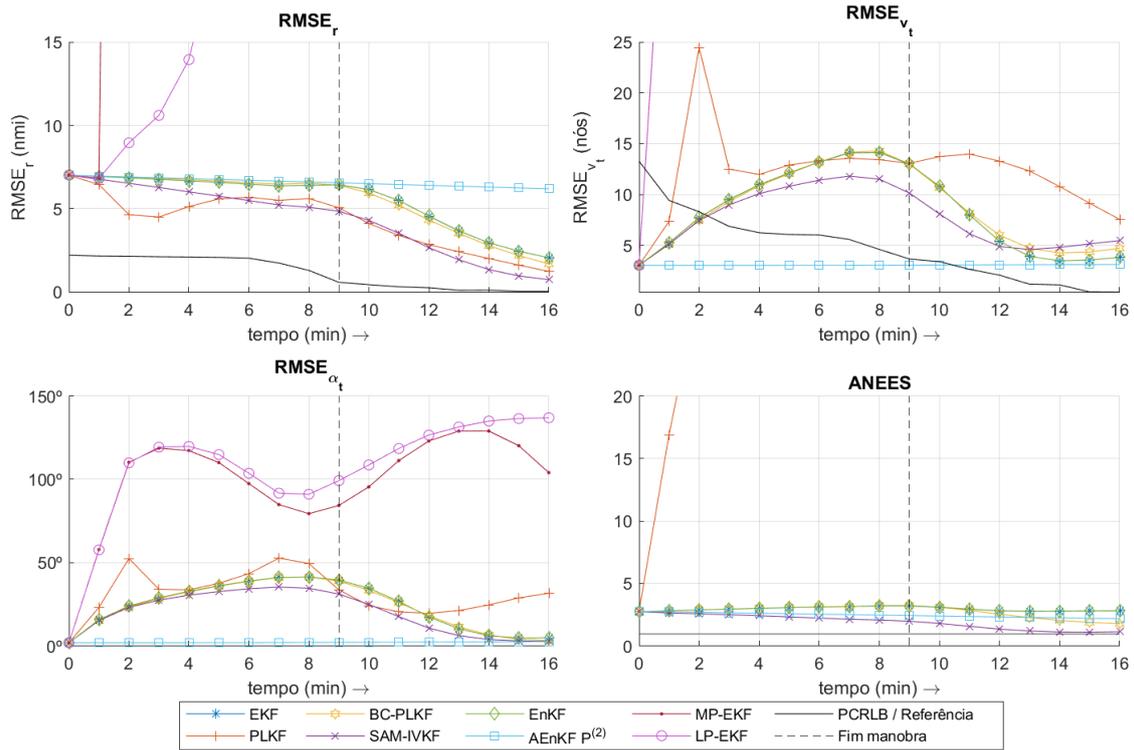


Figura C.3: Métricas –  $M_1I_3$

## C.4 Movimentação 1, Inicialização 4

Tabela C.4: Métricas –  $M_1I_4$  (médias temporais)

Modelo	$\overline{RMSE}_r$	$\overline{RMSE}_{v_t}$	$\overline{RMSE}_{\alpha_t}$	ANEES	$\overline{TPT}$	$\overline{LPT}$	$\overline{RPT}$	SR	TCR
EKF	10,959	43,605	163, 21°	9,525	0,212	0,013	1,000	100%	100%
PLKF	<b>1,863</b>	<b>17,270</b>	154, 45°	157,614	0,131	0,008	0,617	100%	100%
BC-PLKF	<b>10,069</b>	40,448	162, 07°	<b>7,250</b>	0,310	0,019	1,464	100%	100%
SAM-IVKF	<b>9,044</b>	38,994	163, 14°	<b>6,135</b>	0,402	0,025	1,899	100%	100%
EnKF	10,960	43,620	163, 20°	<b>9,515</b>	3,687	0,230	17,427	100%	100%
AEnKF ( $P^{(2)}$ )	11,732	24,821	<b>125, 53°</b>	20,311	14,431	0,902	68,209	100%	100%
MP-EKF	11,970	<b>22,574</b>	<b>138, 13°</b>	180,780	0,341	0,021	1,612	100%	100%
LP-EKF	11,359	<b>18,770</b>	<b>133, 76°</b>	99,316	0,141	0,009	0,665	100%	100%
PCRLB	0,309	2,413							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados **em negrito**

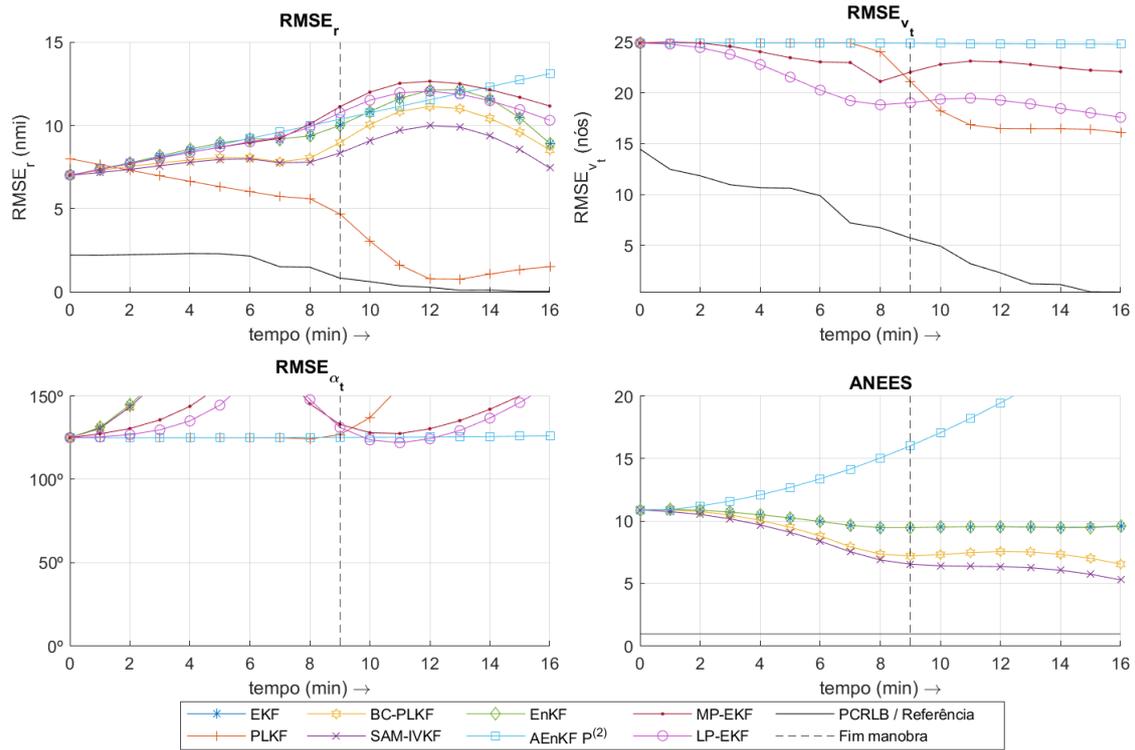


Figura C.4: Métricas –  $M_1I_4$

## C.5 Movimentação 1, Inicialização 5

Tabela C.5: Métricas –  $M_1I_5$  (médias temporais)

Modelo	$\overline{RMSE}_r$	$\overline{RMSE}_{v_t}$	$\overline{RMSE}_{\alpha_t}$	$\overline{ANEES}$	$\overline{TPT}$	$\overline{LPT}$	$\overline{RPT}$	SR	TCR
RP-EKF	6,215	7,265	25, 72°	<b>1,571</b>	10,774	0,673	1,000	100%	100%
RP-PLKF	<b>1,150</b>	<b>2,757</b>	<b>9, 04°</b>	29,832	8,296	0,518	0,770	100%	100%
RP-BC-PLKF	<b>6,065</b>	7,183	25, 77°	<b>1,503</b>	13,244	0,828	1,229	100%	100%
RP-SAM-IVKF	6,069	<b>7,179</b>	25, 73°	<b>1,501</b>	16,797	1,050	1,559	100%	100%
RP-EnKF	6,222	7,262	<b>25, 72°</b>	1,572	137,895	8,618	12,799	100%	100%
RP-AEnKF ( $P^{(2)}$ )	<b>3,305</b>	<b>6,555</b>	<b>10, 81°</b>	2,427	618,118	38,632	57,373	100%	99,95%
RP-MP-EKF	155,275	$1,96 \cdot 10^3$	107, 60°	7,167	14,173	0,886	1,315	100%	98,00%
RP-LP-EKF	7,541	70,736	102, 76°	8,945	9,006	0,563	0,836	100%	99,95%
PCRLB	0,568	1,304							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados **em negrito**

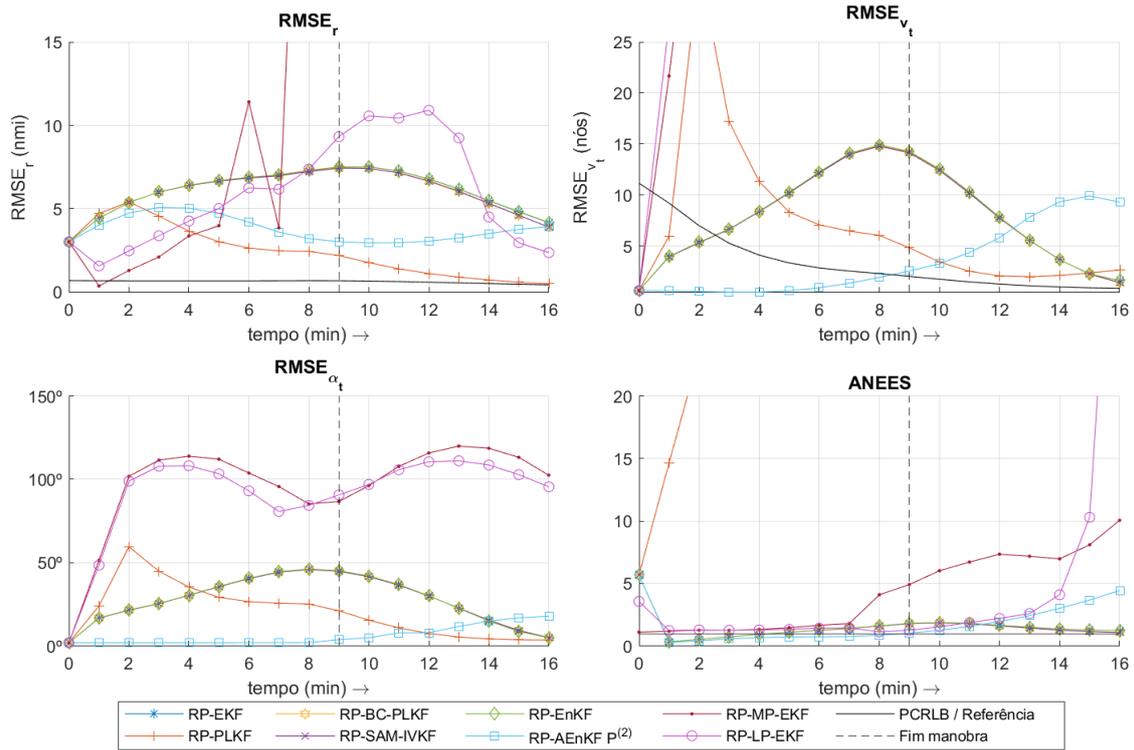


Figura C.5: Métricas –  $M_1I_5$

## C.6 Movimentação 2, Inicialização 1

Tabela C.6: Métricas –  $M_2I_1$  (médias temporais)

Modelo	$\overline{RMSE}_r$	$\overline{RMSE}_{v_t}$	$\overline{RMSE}_{\alpha_t}$	$\overline{ANEES}$	$\overline{TPT}$	$\overline{LPT}$	$\overline{RPT}$	SR	TCR
EKF	<b>2,097</b>	<b>6,203</b>	<b>17, 70°</b>	$2,02 \cdot 10^3$	0,191	0,010	1,000	100%	100%
PLKF	6,930	15,368	63, 55°	<b>204,460</b>	0,113	0,006	0,593	100%	100%
BC-PLKF	<b>2,586</b>	<b>6,380</b>	<b>17, 06°</b>	<b>7,180</b>	0,284	0,014	1,484	100%	99,95%
SAM-IVKF	32,950	81,456	37, 20°	429,932	0,343	0,017	1,792	100%	96,10%
EnKF	<b>2,034</b>	<b>5,646</b>	<b>16, 90°</b>	$1,26 \cdot 10^3$	3,333	0,167	17,429	100%	100%
AEnKF ( $P^{(2)}$ )	3,844	15,630	32, 33°	<b>56,258</b>	13,571	0,679	70,966	100%	99,75%
MP-EKF	197,629	677,618	37, 38°	$1,31 \cdot 10^5$	0,312	0,016	1,630	100%	96,65%
LP-EKF	$1,49 \cdot 10^7$	$1,02 \cdot 10^9$	50, 01°	$1,31 \cdot 10^6$	0,120	0,006	0,629	100%	95,45%
PCRLB	1,276	3,289							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados **em negrito**

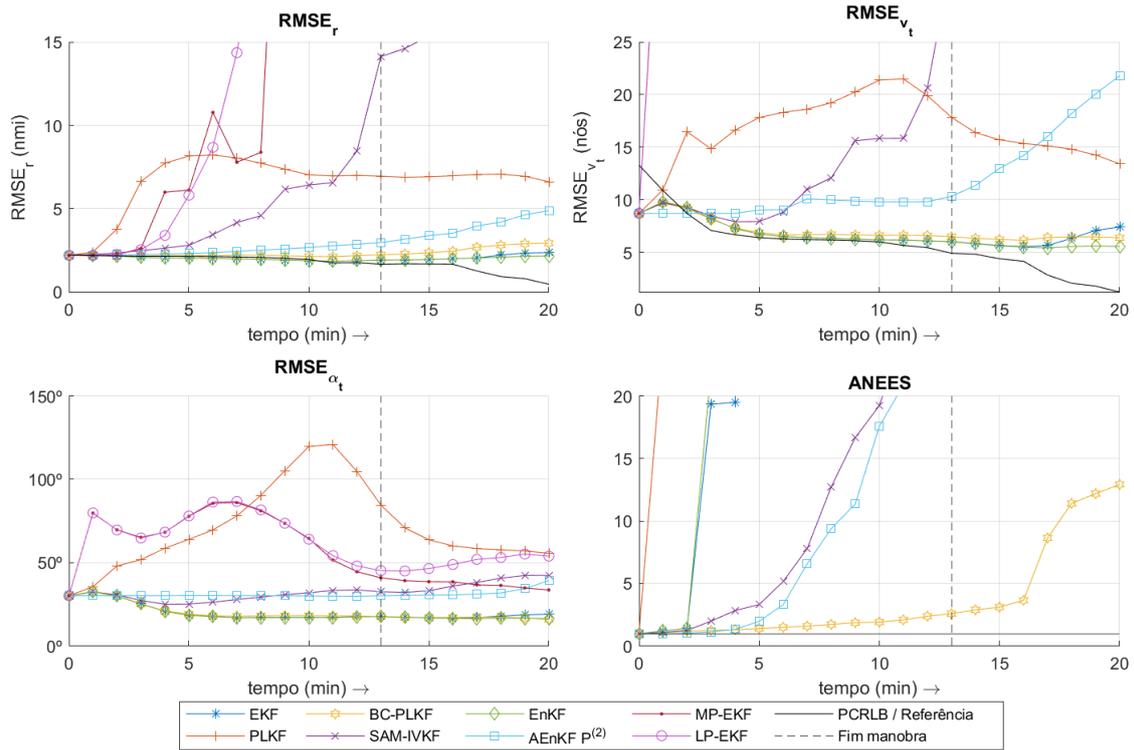


Figura C.6: Métricas –  $M_2I_1$

## C.7 Movimentação 2, Inicialização 2

Tabela C.7: Métricas –  $M_2I_2$  (médias temporais)

Modelo	$\overline{RMSE}_r$	$\overline{RMSE}_{v_t}$	$\overline{RMSE}_{\alpha_t}$	$\overline{ANEES}$	$\overline{TPT}$	$\overline{LPT}$	$\overline{RPT}$	SR	TCR
EKF	<b>4,867</b>	<b>10,851</b>	<b>19,81°</b>	<b>2,199</b>	0,251	0,013	1,000	100%	100%
PLKF	6,864	15,218	62,78°	201,312	0,150	0,007	0,596	100%	100%
BC-PLKF	<b>5,057</b>	11,310	20,28°	<b>3,540</b>	0,364	0,018	1,452	100%	100%
SAM-IVKF	6,309	14,725	<b>16,31°</b>	8,892	0,455	0,023	1,813	100%	99,60%
EnKF	<b>4,874</b>	<b>10,863</b>	<b>19,84°</b>	<b>2,197</b>	4,401	0,220	17,544	100%	100%
AEnKF ( $P^{(2)}$ )	5,356	<b>8,652</b>	28,62°	4,961	17,392	0,870	69,324	100%	100%
MP-EKF	308,836	$1,11 \cdot 10^3$	38,03°	$8,55 \cdot 10^5$	0,405	0,020	1,614	100%	96,85%
LP-EKF	$3,43 \cdot 10^3$	$5,17 \cdot 10^4$	46,86°	$5,73 \cdot 10^6$	0,162	0,008	0,648	100%	94,25%
PCRLB	1,300	3,385							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados **em negrito**

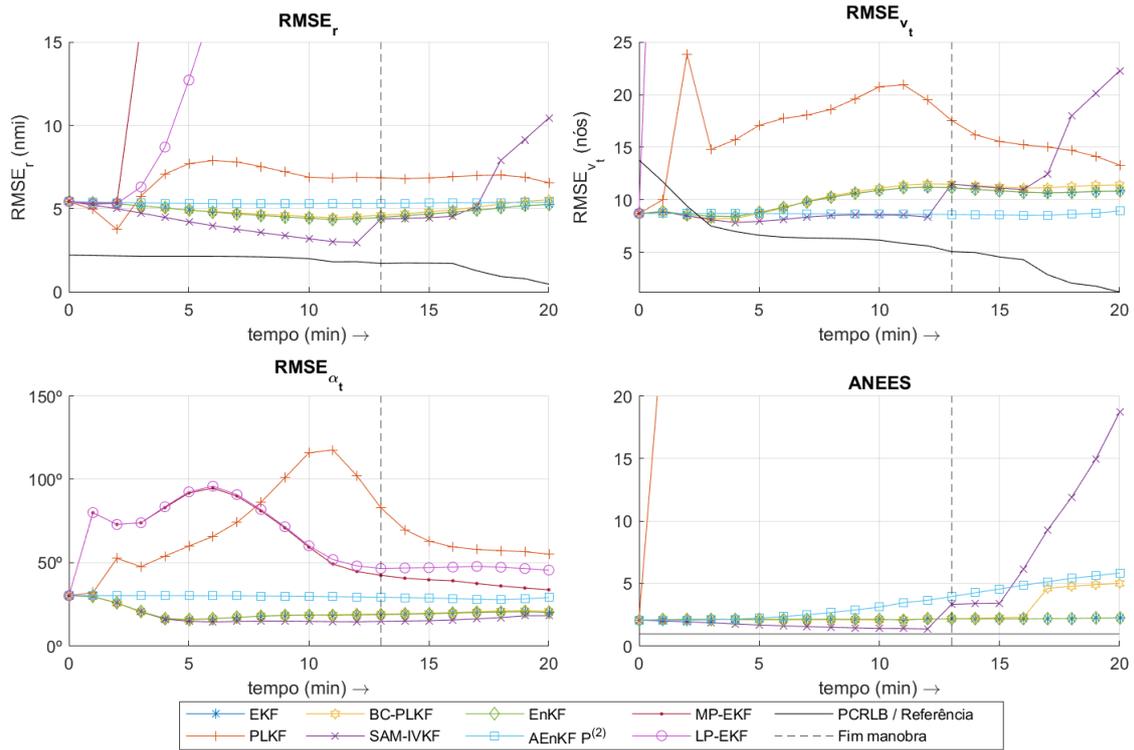


Figura C.7: Métricas –  $M_2I_2$

## C.8 Movimentação 2, Inicialização 3

Tabela C.8: Métricas –  $M_2I_3$  (médias temporais)

Modelo	$\overline{RMSE}_r$	$\overline{RMSE}_{v_t}$	$\overline{RMSE}_{\alpha_t}$	$\overline{ANEES}$	$\overline{TPT}$	$\overline{LPT}$	$\overline{RPT}$	SR	TCR
EKF	<b>4,696</b>	<b>10,292</b>	<b>19,94°</b>	<b>1,801</b>	0,198	0,010	1,000	100%	100%
PLKF	6,867	15,223	62,79°	201,256	0,116	0,006	0,586	100%	100%
BC-PLKF	4,948	10,852	20,32°	2,032	0,290	0,015	1,469	100%	100%
SAM-IVKF	<b>1,978</b>	<b>5,124</b>	<b>9,62°</b>	<b>0,823</b>	0,358	0,018	1,815	100%	100%
EnKF	<b>4,704</b>	10,311	<b>19,99°</b>	<b>1,804</b>	3,410	0,170	17,263	100%	100%
AEnKF ( $P^{(2)}$ )	4,836	<b>7,608</b>	27,82°	4,188	13,942	0,697	70,585	100%	100%
MP-EKF	337,077	$1,37 \cdot 10^3$	38,13°	$1,26 \cdot 10^7$	0,319	0,016	1,613	100%	97,20%
LP-EKF	$1,25 \cdot 10^3$	$9,65 \cdot 10^4$	44,29°	$7,70 \cdot 10^5$	0,123	0,006	0,622	100%	96,00%
PCRLB	1,635	4,000							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados **em negrito**

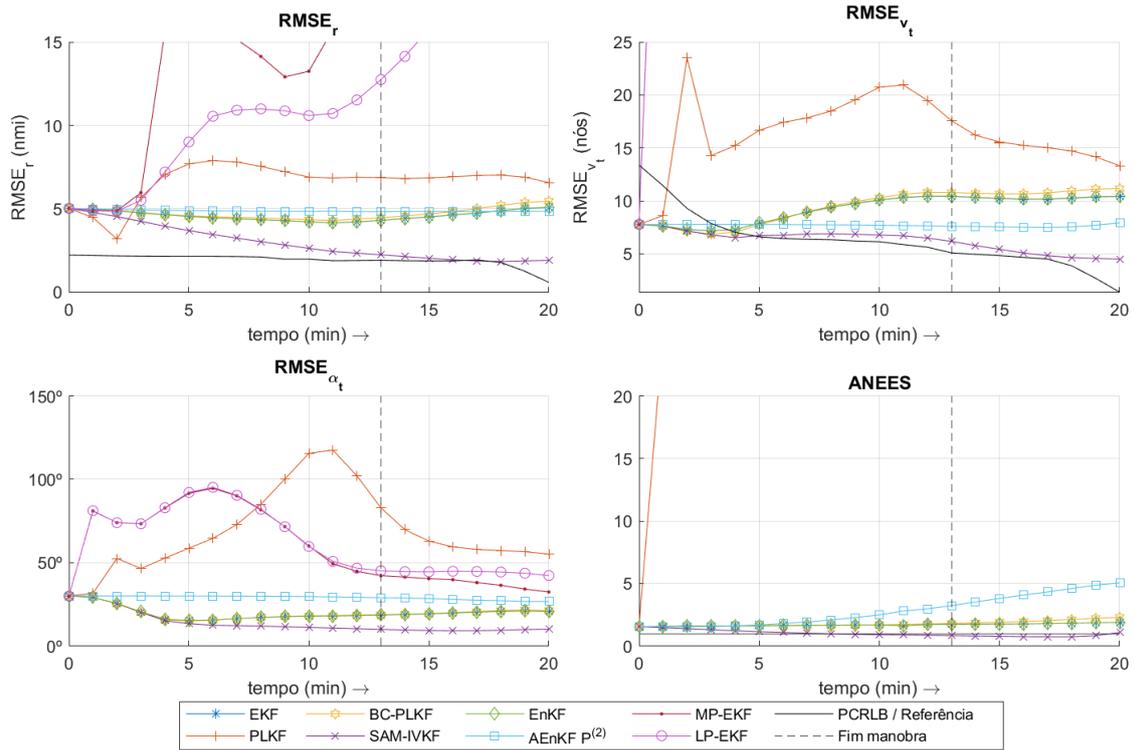


Figura C.8: Métricas –  $M_2I_3$

## C.9 Movimentação 2, Inicialização 4

Tabela C.9: Métricas –  $M_2I_4$  (médias temporais)

Modelo	$\overline{RMSE}_r$	$\overline{RMSE}_{v_t}$	$\overline{RMSE}_{\alpha_t}$	ANEES	$\overline{TPT}$	$\overline{LPT}$	$\overline{RPT}$	SR	TCR
EKF	<b>1,008</b>	<b>8,671</b>	<b>25,16°</b>	<b>14,142</b>	0,196	0,010	1,000	100%	100%
PLKF	6,973	16,994	85,34°	340,724	0,118	0,006	0,602	100%	100%
BC-PLKF	<b>1,055</b>	<b>8,489</b>	<b>23,15°</b>	<b>13,072</b>	0,294	0,015	1,499	100%	100%
SAM-IVKF	2,001	9,785	26,00°	26,770	0,359	0,018	1,830	100%	99,95%
EnKF	<b>1,012</b>	<b>8,657</b>	<b>25,20°</b>	<b>13,761</b>	3,419	0,171	17,429	100%	100%
AEnKF ( $P^{(2)}$ )	11,515	26,856	153,09°	53,593	14,066	0,703	71,696	100%	100%
MP-EKF	13,121	26,375	52,37°	74,660	0,323	0,016	1,648	100%	100%
LP-EKF	12,964	23,198	46,57°	41,909	0,125	0,006	0,635	100%	100%
PCRLB	0,858	2,010							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados **em negrito**

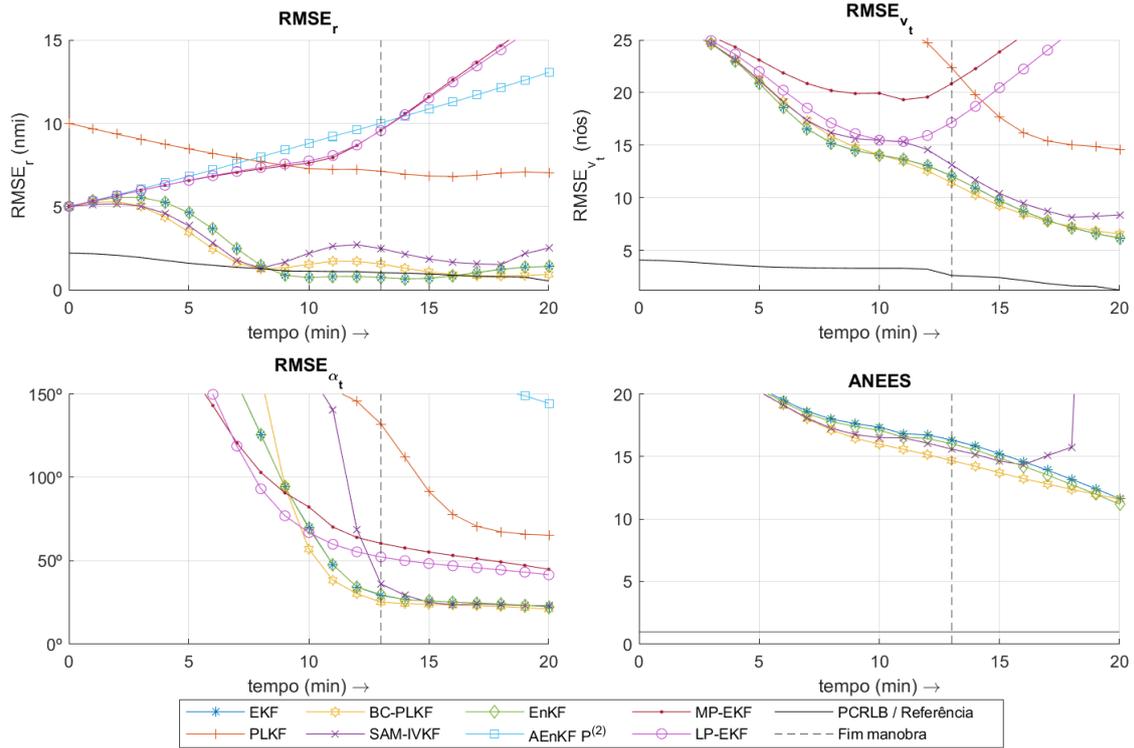


Figura C.9: Métricas – M<sub>2</sub>I<sub>4</sub>

## C.10 Movimentação 2, Inicialização 5

Tabela C.10: Métricas – M<sub>2</sub>I<sub>5</sub> (médias temporais)

Modelo	RMSE <sub>r</sub>	RMSE <sub>v<sub>t</sub></sub>	RMSE <sub>α<sub>t</sub></sub>	ANEES	TPT	LPT	RPT	SR	TCR
RP-EKF	4,468	9,657	<b>19,37°</b>	<b>0,716</b>	13,146	0,657	1,000	100%	100%
RP-PLKF	4,821	10,834	44,68°	132,309	8,195	0,410	0,623	100%	100%
RP-BC-PLKF	4,413	<b>9,563</b>	<b>19,12°</b>	<b>0,711</b>	16,321	0,816	1,242	100%	100%
RP-SAM-IVKF	<b>4,301</b>	<b>9,317</b>	<b>18,69°</b>	0,684	20,435	1,022	1,554	100%	100%
RP-EnKF	4,476	9,672	19,40°	<b>0,713</b>	165,048	8,252	12,555	100%	100%
RP-AEnKF (P <sup>(2)</sup> )	<b>2,132</b>	<b>8,712</b>	21,96°	3,878	661,567	33,078	50,324	100%	100%
RP-MP-EKF	82,840	241,952	35,42°	3,397	17,346	0,867	1,319	100%	98,95%
RP-LP-EKF	<b>2,598</b>	22,105	44,85°	2,073	11,000	0,550	0,837	100%	100%
PCRLB	0,804	1,982							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados **em negrito**

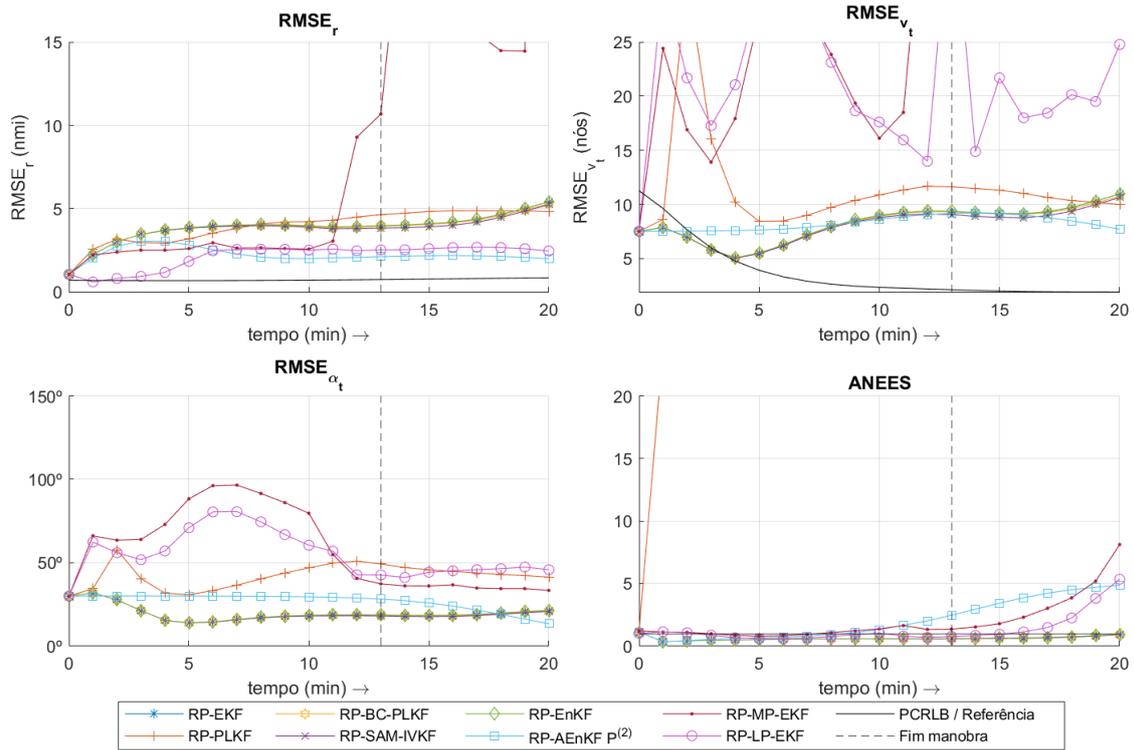


Figura C.10: Métricas –  $M_2I_5$

## C.11 Movimentação 3, Inicialização 1

Tabela C.11: Métricas –  $M_3I_1$  (médias temporais)

Modelo	$\overline{RMSE}_r$	$\overline{RMSE}_{v_t}$	$\overline{RMSE}_{\alpha_t}$	$\overline{ANEES}$	$\overline{TPT}$	$\overline{LPT}$	$\overline{RPT}$	SR	TCR
EKF	38,470	98,021	35,05°	$1,13 \cdot 10^3$	0,264	0,009	1,000	100%	51,90%
PLKF	7,146	15,535	108,27°	349,482	0,160	0,006	0,604	100%	100%
BC-PLKF	<b>2,201</b>	<b>4,758</b>	<b>20,03°</b>	<b>3,364</b>	0,394	0,014	1,491	100%	100%
SAM-IVKF	<b>3,968</b>	<b>7,151</b>	<b>30,62°</b>	<b>28,095</b>	0,485	0,017	1,837	100%	99,75%
EnKF	<b>1,712</b>	<b>4,151</b>	<b>16,84°</b>	<b>2,508</b>	4,586	0,164	17,367	100%	100%
AEnKF ( $P^{(2)}$ )	62,605	256,036	72,31°	$5,58 \cdot 10^3$	18,813	0,672	71,244	100%	19,40%
MP-EKF	162,490	736,035	92,46°	$1,33 \cdot 10^{10}$	0,432	0,015	1,635	100%	97,85%
LP-EKF	$6,55 \cdot 10^4$	$6,55 \cdot 10^4$	85,40°	$1,33 \cdot 10^{113}$	0,167	0,006	0,632	100%	12,85%
PCRLB	1,798	4,095							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados em **negrito**

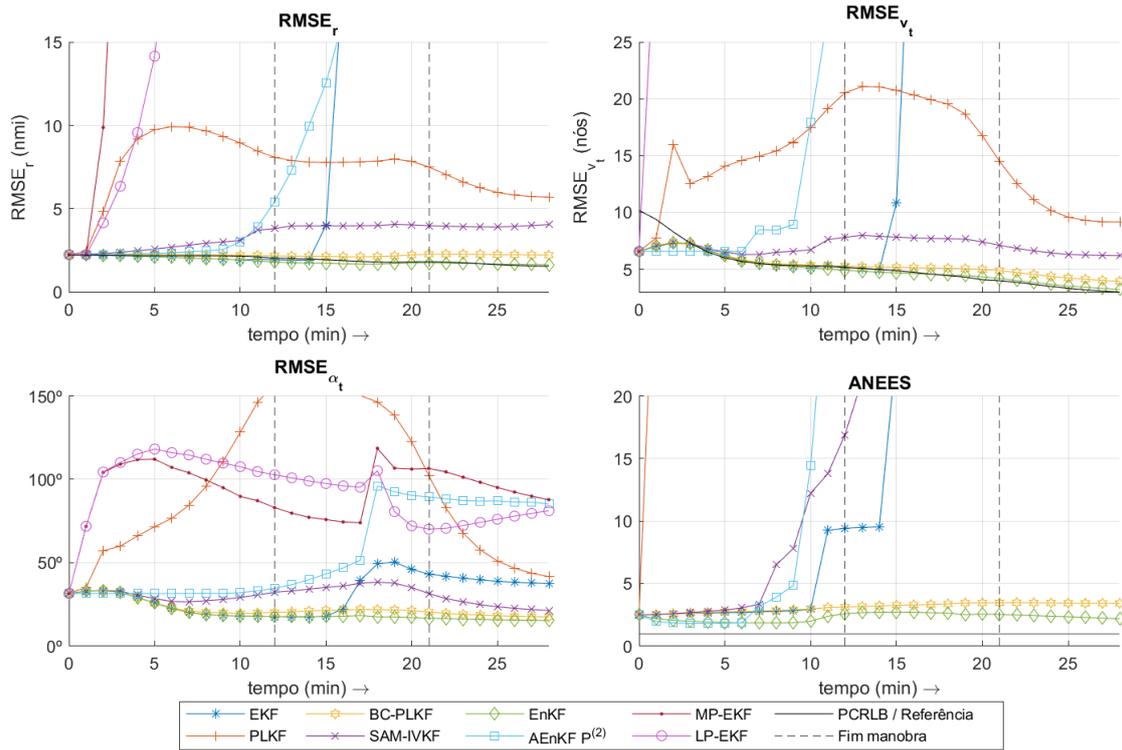


Figura C.11: Métricas –  $M_3I_1$

## C.12 Movimentação 3, Inicialização 2

Tabela C.12: Métricas –  $M_3I_2$  (médias temporais)

Modelo	$\overline{RMSE}_r$	$\overline{RMSE}_{v_t}$	$\overline{RMSE}_{\alpha_t}$	$\overline{ANEES}$	$\overline{TPT}$	$\overline{LPT}$	$\overline{RPT}$	SR	TCR
EKF	41,983	106,917	29,43°	$1,12 \cdot 10^3$	0,266	0,010	1,000	100%	47,20%
PLKF	7,151	15,318	106,72°	344,866	0,160	0,006	0,601	100%	100%
BC-PLKF	<b>3,179</b>	<b>7,565</b>	<b>15,76°</b>	<b>2,220</b>	0,396	0,014	1,486	100%	100%
SAM-IVKF	<b>2,224</b>	<b>6,626</b>	<b>20,20°</b>	<b>5,299</b>	0,487	0,017	1,829	100%	100%
EnKF	<b>2,808</b>	<b>7,051</b>	<b>13,33°</b>	<b>2,035</b>	4,645	0,166	17,449	100%	100%
AEnKF ( $P^{(2)}$ )	61,563	246,825	69,99°	$3,30 \cdot 10^3$	19,037	0,680	71,503	100%	15,35%
MP-EKF	—	—	—	—	0,433	0,015	1,626	0%	0%
LP-EKF	—	—	—	—	0,168	0,006	0,632	0%	0%
PCRLB	1,874	4,232							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados **em negrito**

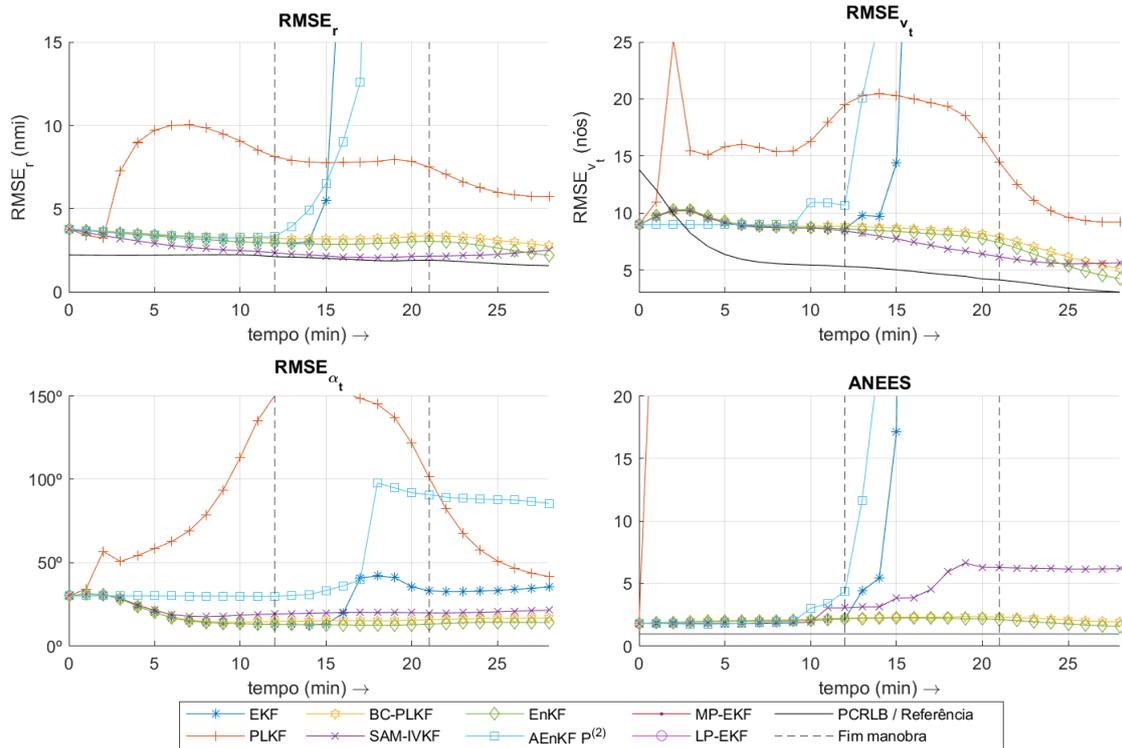


Figura C.12: Métricas –  $M_3I_2$

### C.13 Movimentação 3, Inicialização 3

Tabela C.13: Métricas –  $M_3I_3$  (médias temporais)

Modelo	$\overline{RMSE}_r$	$\overline{RMSE}_{v_t}$	$\overline{RMSE}_{\alpha_t}$	$\overline{ANEES}$	$\overline{TPT}$	$\overline{LPT}$	$\overline{RPT}$	SR	TCR
EKF	36,948	97,585	24,09°	999,710	0,270	0,010	1,000	100%	50,60%
PLKF	7,153	15,316	106,72°	344,870	0,162	0,006	0,601	100%	100%
BC-PLKF	<b>2,758</b>	<b>6,751</b>	<b>12,59°</b>	<b>1,775</b>	0,400	0,014	1,482	100%	100%
SAM-IVKF	<b>1,398</b>	<b>5,656</b>	<b>16,89°</b>	<b>1,306</b>	0,492	0,018	1,821	100%	100%
EnKF	<b>2,262</b>	<b>6,174</b>	<b>9,69°</b>	<b>1,515</b>	4,672	0,167	17,314	100%	100%
AEnKF ( $P^{(2)}$ )	61,652	242,518	70,63°	$3,38 \cdot 10^3$	19,203	0,686	71,159	100%	10,45%
MP-EKF	—	—	—	—	0,438	0,016	1,624	0%	0%
LP-EKF	—	—	—	—	0,170	0,006	0,631	0%	0%
PCRLB	1,901	4,268							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados **em negrito**

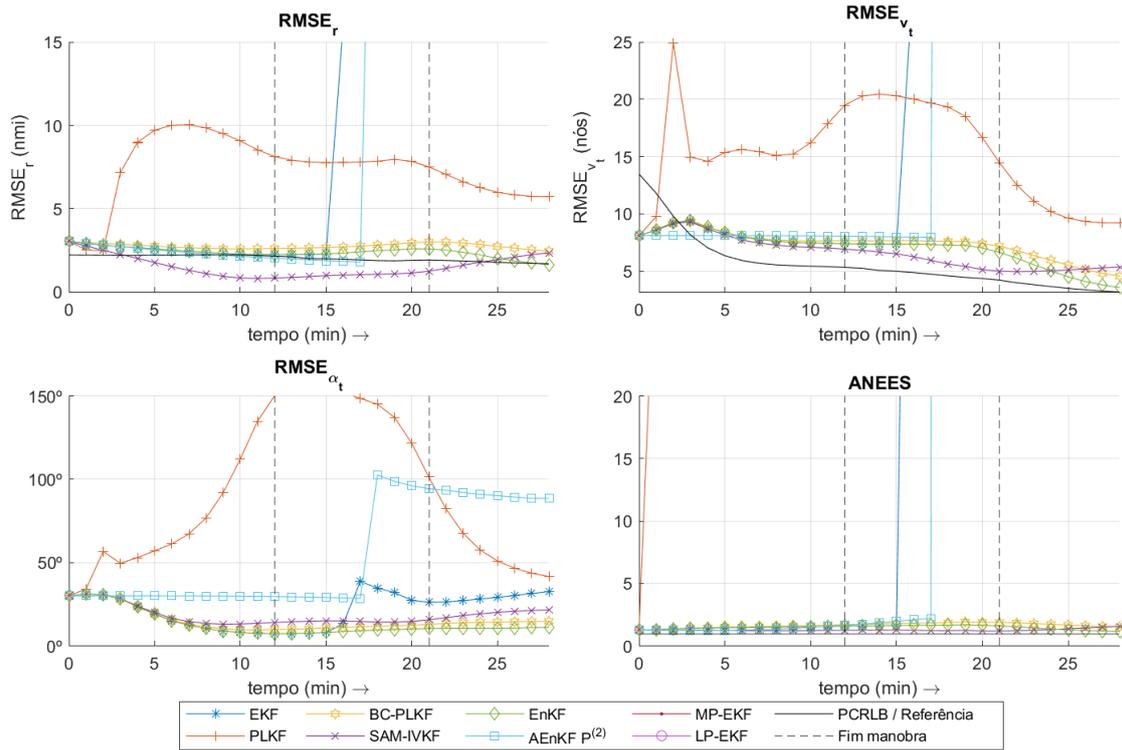


Figura C.13: Métricas –  $M_3I_3$

## C.14 Movimentação 3, Inicialização 4

Tabela C.14: Métricas –  $M_3I_4$  (médias temporais)

Modelo	$\overline{RMSE}_r$	$\overline{RMSE}_{v_t}$	$\overline{RMSE}_{\alpha_t}$	ANEES	TPT	LPT	RPT	SR	TCR
EKF	120,913	242,863	110,56°	$2,96 \cdot 10^3$	0,283	0,010	1,000	100%	4,75%
PLKF	7,203	<b>17,238</b>	119,90°	486,790	0,171	0,006	0,606	100%	100%
BC-PLKF	<b>5,313</b>	<b>20,180</b>	92,61°	<b>4,895</b>	0,421	0,015	1,490	100%	100%
SAM-IVKF	<b>3,159</b>	<b>16,923</b>	<b>91,84°</b>	<b>4,738</b>	0,521	0,019	1,844	100%	100%
EnKF	<b>5,585</b>	22,468	100,66°	<b>6,633</b>	4,917	0,176	17,387	100%	100%
AEnKF ( $P^{(2)}$ )	64,718	249,690	124,39°	$3,48 \cdot 10^3$	20,169	0,720	71,312	100%	11,25%
MP-EKF	10,375	44,871	<b>64,11°</b>	$2,26 \cdot 10^5$	0,463	0,017	1,639	100%	100%
LP-EKF	297,821	$1,77 \cdot 10^3$	<b>65,86°</b>	$8,14 \cdot 10^4$	0,182	0,006	0,643	100%	0%
PCRLB	1,897	4,621							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados **em negrito**

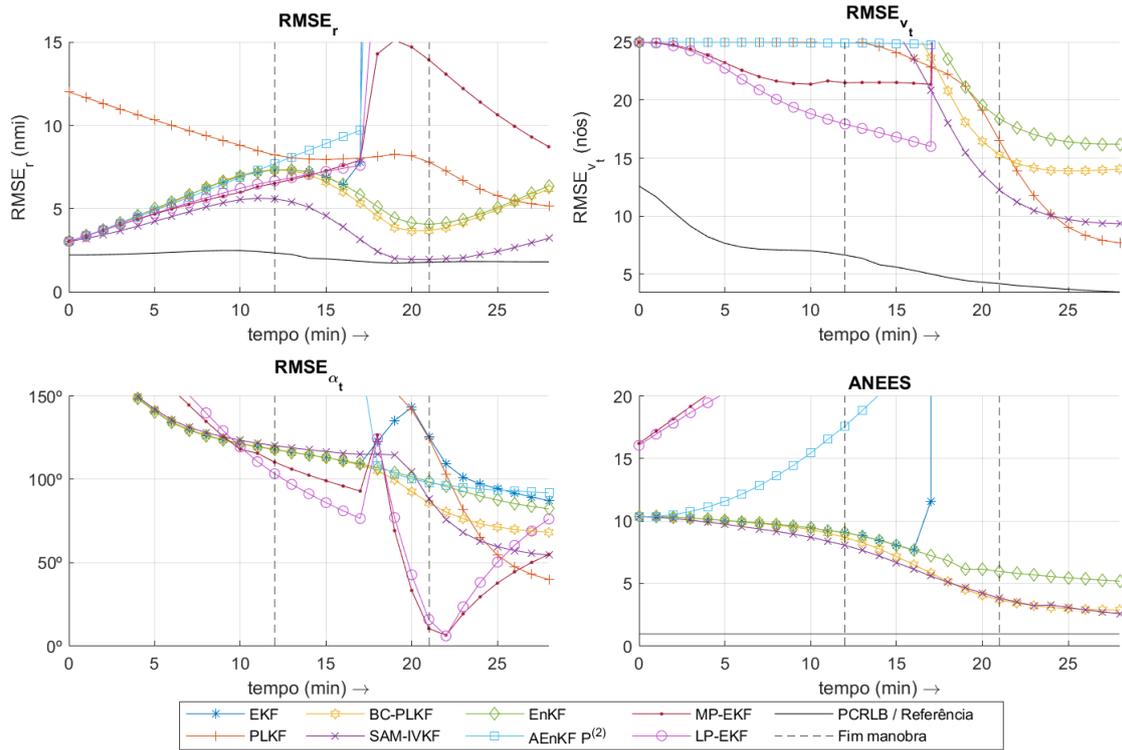


Figura C.14: Métricas – M<sub>3</sub>I<sub>4</sub>

## C.15 Movimentação 3, Inicialização 5

Tabela C.15: Métricas – M<sub>3</sub>I<sub>5</sub> (médias temporais)

Modelo	$\overline{\text{RMSE}}_r$	$\overline{\text{RMSE}}_{v_t}$	$\overline{\text{RMSE}}_{\alpha_t}$	ANEES	$\overline{\text{TPT}}$	$\overline{\text{LPT}}$	$\overline{\text{RPT}}$	SR	TCR
RP-EKF	3,615	7,681	14,95°	2,378	13,004	0,464	1,000	100%	99,90%
RP-PLKF	5,769	10,322	65,47°	237,972	8,662	0,309	0,666	100%	100%
RP-BC-PLKF	<b>2,777</b>	<b>6,184</b>	<b>9,84°</b>	<b>0,977</b>	19,783	0,707	1,521	100%	100%
RP-SAM-IVKF	<b>2,638</b>	<b>6,026</b>	<b>9,37°</b>	<b>0,957</b>	25,379	0,906	1,952	100%	100%
RP-EnKF	<b>2,772</b>	<b>6,174</b>	<b>9,59°</b>	<b>0,964</b>	194,597	6,950	14,965	100%	100%
RP-AEnKF (P <sup>(2)</sup> )	33,310	125,881	67,88°	1,20 · 10 <sup>3</sup>	658,354	23,513	50,629	100%	48,90%
RP-MP-EKF	140,546	1,34 · 10 <sup>3</sup>	85,23°	2,78 · 10 <sup>14</sup>	14,938	0,533	1,149	100%	97,75%
RP-LP-EKF	–	–	–	–	9,580	0,342	0,737	0%	0%
PCRLB	0,784	1,652							

Obs.: Distância em milhas, velocidade em nós e tempo em milissegundos. Melhores resultados **em negrito**

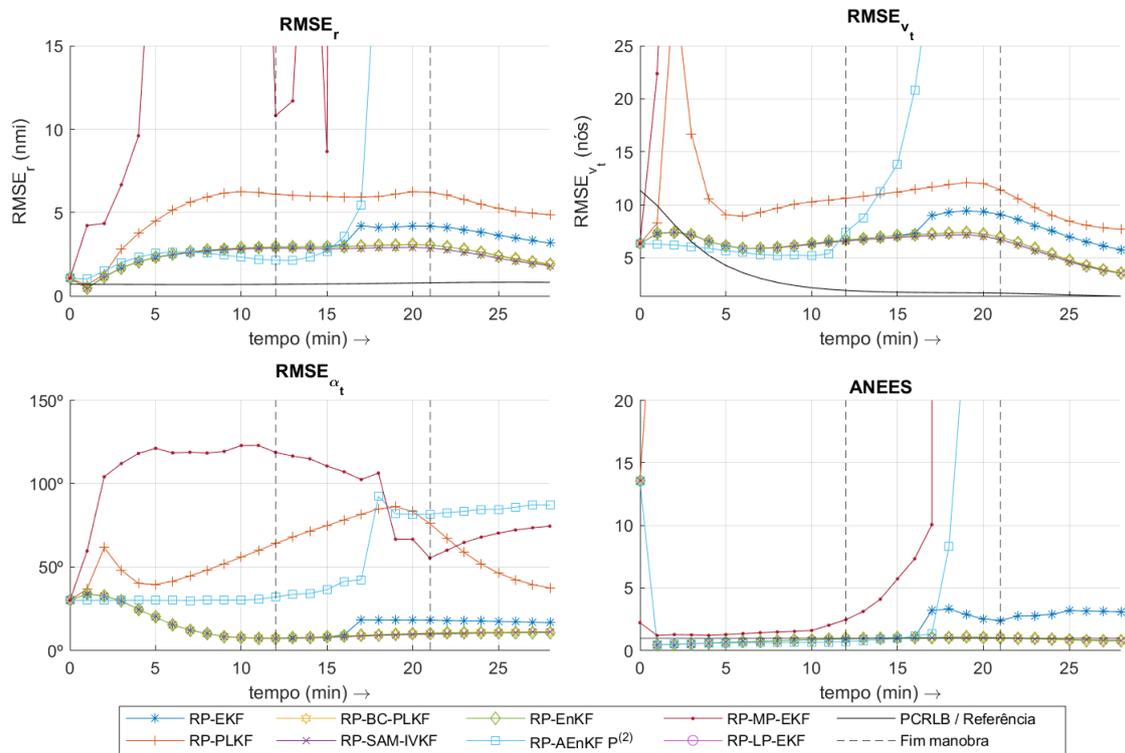


Figura C.15: Métricas – M<sub>3</sub>I<sub>5</sub>