Leonardo Azevedo Scardua

Sintonia Automática do Filtro de Kalman Unscented

São Paulo
2015

Leonardo Azevedo Scardua

Sintonia Automática do Filtro de Kalman Unscented

Tese apresentada à Escola Politécnica
da Universidade de São Paulo para
obtenção do título de Doutor

São Paulo
2015

Leonardo Azevedo Scardua

Sintonia Automática do Filtro de Kalman Unscented

Tese apresentada à Escola Politécnica
da Universidade de São Paulo para
obtenção do título de Doutor

Área de Concentração:
Engenharia de Sistemas

Orientador:
Prof. Dr. José Jaime da Cruz

São Paulo
2015

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Ficha Catalográfica

Scardua, Leonardo

Sintonia Automática do Filtro de Kalman Unscented / L. Scardua -- versão corr. -- São Paulo, 2015.

95 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Telecomunicações e Controle.

1.Estimação não linear 2.Filtro de Kalman Unscented I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Telecomunicações e Controle II.t.

Dedicatória

À Angelita.
Aos meus pais.
Ao meu irmão.

Resumo

SCARDUA, L. A. **Sintonia Automática do Filtro de Kalman Unscented**. 2015. 95 f. Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2015.

O filtro de Kalman estendido tem sido a mais popular ferramenta de filtragem não linear das últimas quatro décadas. É de fácil implementação e apresenta baixo custo computacional. Nos casos nos quais as não linearidades do sistema dinâmico são significativas, porém, o filtro de Kalman estendido pode apresentar resultados insatisfatórios. Nessas situações, o filtro de Kalman *unscented* substitui com vantagens o filtro de Kalman estendido, pois pode apresentar melhores estimativas de estado, embora ambos os filtros exibam complexidade computacional de mesma ordem.

A qualidade das estimativas de estado do filtro *unscented* está intimamente ligada à sintonia dos parâmetros que controlam a transformada *unscented*. A versão escalada dessa transformada exibe três parâmetros escalares que determinam o posicionamento dos pontos sigma e, consequentemente, afetam diretamente a qualidade das estimativas produzidas pelo filtro. Apesar da importância do filtro de Kalman *unscented*, a sintonia ótima desses parâmetros é um problema para o qual ainda não há solução definitiva. Não há nem mesmo recomendações heurísticas que garantam o bom funcionamento do filtro *unscented* na maior parte dos problemas tratáveis por meio de filtros Gaussianos. Essa carência e a importância desse filtro para a área de filtragem não linear fazem da busca por mecanismos de sintonia automática do filtro *unscented* área de pesquisa ativa. Assim, este trabalho propõe técnicas para sintonia automática dos parâmetros da transformada *unscented* escalada. Além da sintonia desses parâmetros, também é abordado o problema de sintonizar as matrizes de covariância dos ruídos de processo e de medida demandadas pelo modelo do sistema dinâmico usado pelo filtro *unscented*. As técnicas propostas cobrem então a sintonia automática de todos os parâmetros do filtro.

Palavras-chave: estimação não linear, filtro de Kalman *unscented*.

Abstract

SCARDUA, L. A. **Automatic Tuning of the Unscented Kalman Filter**. 2015. 95 f. Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2015.

The extended Kalman filter has been the most popular nonlinear filter of the last four decades. It is easy to implement and exhibits low computational cost. When nonlinearities are significant, though, the extended Kalman filter can display poor state estimation performance. In such situations, the unscented Kalman filter can yield better state estimates, while displaying the same order of computational complexity as the extended Kalman filter.

The quality of the state estimates produced by the unscented Kalman filter is directly influenced by the tuning of the scalar parameters that govern the unscented transform. The scaled version of the unscented transform features three scalar parameters that determine the positioning of the sigma points, thus directly affecting the filter state estimation performance. Despite the importance of the unscented Kalman filter, the optimal tuning of the scaled unscented transform parameters is still an open problem. This work hence discusses algorithms for the automatic tuning of the unscented transform parameters. The discussion includes the tuning of the needed noise covariance matrices, thus covering the automatic tuning of all parameters of the unscented Kalman filter.

Keywords: nonlinear estimation, unscented Kalman filter

Lista de Figuras

Lista de Tabelas

Lista de Siglas

B

BSF Filtro de Partículas de *Bootstrap*. 1, 22, 23

C

CKF Filtro de Kalman de Cubatura. 9, 32, 40, 47, 50, 60, 71

E

EI *Expected Improvement*. 27, 29, 30, 37, 41, 44, 48, 51, 54

EKF Filtro de Kalman Estendido. 6, 7, 9–11, 15, 19, 31, 40, 47, 50, 60, 65, 68, 71

G

GHKF Filtro de Kalman de Gauss-Hermite. 9

GP Processos Gaussianos. 25, 26, 28–30

K

KF Filtro de Kalman. 5–7, 15

M

MSE Erro Médio Quadrático. 32, 33, 35

MSEx Erro Médio Quadrático nas Estimativas de Estado. 33, 38–40, 43, 46, 49, 50, 53, 56–58, 60, 62, 64

MSEy Erro Médio Quadrático nas Estimativas de Medida. 33, 35, 40, 43, 46, 50, 53, 56

N

NCI Índice de não-confiabilidade. 32, 34, 38–40, 49, 57, 64

NLL Negativo do Logaritmo da Verossimilhança. 32

NLLy Negativo do Logaritmo da Verossimilhança das predições de medida. 35, 40, 43, 46, 50, 53, 56

P

PF Filtro de Partículas. 6, 7

PfUkf Filtro de Kalman *unscented* sintonizado pelo Algoritmo 3. 31, 32, 36, 37, 40, 41, 43, 44, 46, 60, 62, 64, 68, 71

PfUkf-QR PfUkf que inclui a sintonia das matrizes de covariância dos ruídos de processo e de medida. 47, 48, 50, 53, 56, 62

S

SE *Exponencial Quadrática*. 25

SMBO Otimização Sequencial Baseada em Modelos. 26, 27

U

UCB *Upper Confidence Bound*. 26, 29, 30, 68

UKF Filtro de Kalman *Unscented*. 1, 6, 7, 9–24, 28–32, 38, 47, 65, 71–73

Ukf-ML Filtro de Kalman *Unscented* sintonizado de acordo com ([1], [2], [3]). 32, 38, 40, 41, 43, 46, 47, 50, 53, 56, 60, 64, 71

UkfA Filtro de Kalman *Unscented* sintonizado pelo Algoritmo 5. 31, 32, 36, 37, 40, 41, 43, 44, 46, 60, 62, 64, 68, 71

UkfA-QR UkfA que inclui a sintonia das matrizes de covariância dos ruídos de processo e de medida. 47, 48, 50, 51, 53, 54, 56, 62

UkfD Filtro de Kalman *Unscented* sintonizado de acordo com [4]. 31, 40, 41, 43, 47, 50, 53, 60

UkfO Filtro de Kalman *unscented* sintonizado de acordo com [5]. 30, 32, 36, 37, 40, 41, 43, 44, 46, 60, 62, 64, 68, 71

UkfO-QR UkfO que inclui a sintonia das matrizes de covariância dos ruídos de processo e de medida. 47, 48, 50, 53, 56, 62

UT Transformada *Unscented* Escalada. 1, 6, 9, 11, 12, 15–19, 31, 32, 41, 47, 51, 65, 71, 73

Sumário

Capítulo 1

Introdução

1.1 Objetivo e Contribuições

Neste trabalho, o tema central é o problema de filtragem ótima. Mais especificamente, propomos algoritmos que abordam a tarefa de encontrar valores para todos os parâmetros do Filtro de Kalman *Unscented* (UKF) [6], [7], [8]), incluindo as matrizes de covariância dos ruídos de processo e de medidas, de modo que as estimativas de estado produzidas pelo filtro sejam melhores que aquelas obtidas por meio da sintonia manual desses parâmetros. Os algoritmos de sintonia propostos tomam como base somente medidas recebidas do sistema dinâmico que será objeto da filtragem, prescindindo do uso de qualquer dado a respeito do estado real do sistema dinâmico. Essa característica torna os algoritmos propostos amplamente aplicáveis, contribuindo para liberar aqueles que utilizam o UKF da necessidade de ajustar manualmente e com quase nenhuma diretriz teórica os vários parâmetros que determinam o funcionamento do filtro. Nesse sentido, as contribuições desse trabalho são:

- A proposição de algoritmo inspirado no Filtro de Partículas de *Bootstrap* (BSF) [9] para a sintonia dos parâmetros da Transformada *Unscented* Escalada (UT) [10] e das matrizes de covariância dos ruídos de processo e de medida;
- A proposição de algoritmo baseado na teoria da otimização por modelos substitutos [11] para a sintonia dos parâmetros da UT e das matrizes de covariância dos ruídos de processo e de medida;
- A aplicação dos algoritmos propostos a um problema complexo, concebido para oferecer desafio significativo a filtros não lineares. O que torna esse problema mais interessante é que ele foi considerado particularmente difícil para o UKF.

1.2 Delineamento da Tese

Nos capítulos 2 a 4, descrevemos o problema de filtragem Bayesiana e, em particular, o UKF. No capítulo 5, apresentamos revisão bibliográfica de trabalhos relevantes a respeito do problema de sintonia do UKF. No capítulo 6, descrevemos a moldura geral da sintonia do UKF como um problema de otimização. As contribuições originais dessa tese são apresentadas e experimentadas nos capítulos 7 a 12. Por fim, as conclusões são apresentadas no capítulo 13.

Capítulo 2

Estimação Bayesiana Sequencial

Neste capítulo, o problema de filtragem ótima é descrito sob a perspectiva da inferência Bayesiana, a qual provê solução ótima para a estimação do estado de um sistema dinâmico [12]. Essa perspectiva estabelece uma moldura probabilística que tem no teorema de Bayes seu principal elemento, permitindo desenvolver algoritmos de filtragem que funcionam de maneira sequencial no tempo.

O funcionamento sequencial evita que a carga computacional do filtro cresça indefinidamente com o recebimento de novas observações, permitindo assim que o filtro seja aplicado em problemas que demandam atualização da estimativa de estado em tempo real. O termo *filtragem Bayesiana* [13] se refere à formulação Bayesiana do problema de filtragem ótima [1]. Assim, deste ponto em diante, usaremos os termos filtragem ótima e filtragem Bayesiana de forma intercambiável, ressaltando que sempre estaremos nos referindo à filtragem (ótima) Bayesiana.

Em termos matemáticos, a filtragem ótima Bayesiana trata de problemas de inversão estatística nos quais a quantidade desconhecida é a sequência de estados do sistema, a qual é observada por meio das medidas ruidosas [14]. A inversão estatística objetiva estimar os estados dadas as observações. Para calcular a distribuição posterior conjunta de todos os estados dadas todas as medidas, basta aplicar diretamente a regra de Bayes. No restante desta seção e na próxima, veremos como isso é feito, de acordo com o material exposto no item 2.5 de [15].

Sejam assim $\{\mathbf{x}_k\}, k = 0, \cdots, N$, um conjunto de vetores $\mathbf{x} \in \Re^{n_x}$ não observáveis e $\{\mathbf{z}_k\}, k = 0, \cdots, N$, o conjunto de medidas $\mathbf{z} \in \Re^{n_z}$ correspondentes. Na estimação Bayesiana recursiva, o objetivo consiste em estimar sequencialmente no tempo a densidade posterior conjunta $p(\mathbf{x}_0, \cdots, \mathbf{x}_N; \mathbf{z}_0, \cdots, \mathbf{z}_N)$. Uma vez determinada, essa densidade pode fornecer várias estatísticas que caracterizam o processo sendo estudado.

Sejam então dois conjuntos de vetores estocásticos $\mathbf{X}^k := \{\mathbf{x}_0, \cdots, \mathbf{x}_k\}$[2] e $\mathbf{Z}^k := \{\mathbf{z}_0, \cdots, \mathbf{z}_k\}$, sendo $\mathbf{X}^k$ o conjunto não observável de variáveis aleatórias que nos interessam e $\mathbf{Z}^k$ o conjunto de medidas do processo estocástico $\mathbf{X}^k$. Segundo a regra de Bayes, a densidade posterior conjunta é dada por

$$p(\mathbf{X}^k|\mathbf{Z}^k) = p(\mathbf{Z}^k|\mathbf{X}^k)\frac{p(\mathbf{X}^k)}{p(\mathbf{Z}^k)} \tag{2.1}$$

onde

$p(\mathbf{X}^k)$ é a densidade conjunta *anterior*,

[1] Ver capítulo 1 de [14].

[2] Assim como em [16], a notação $(\cdot)^k$ denota sequência que inclui o instante k. Por exemplo, $w^k = \{w_0, w_1, \ldots, w_k\}$

$p(\mathbf{Z}^k|\mathbf{X}^k)$ é a *verossimilhança* das observações, e

$p(\mathbf{Z}^k)$ é denominado *evidência*.

De acordo com o *teorema da probabilidade total*, a evidência pode ser reescrita como

$$p(\mathbf{Z}^k) = \int_{\Re^{n_x}} p(\mathbf{Z}^k|\mathbf{X}^k)p(\mathbf{X}^k)d\mathbf{X}^k \ . \tag{2.2}$$

Substituindo 2.2 em 2.1, obtemos a seguinte expressão para a densidade posterior conjunta

$$p(\mathbf{X}^k|\mathbf{Z}^k) = \frac{p(\mathbf{Z}^k|\mathbf{X}^k)p(\mathbf{X}^k)}{\int_{\Re^{n_x}} p(\mathbf{Z}^k|\mathbf{X}^k)p(\mathbf{X}^k)d\mathbf{X}^k} \ . \tag{2.3}$$

Uma vez determinada a distribuição posterior conjunta, as inferências e estimativas desejadas podem ser realizadas por meio de integração. A cada nova observação recebida, porém, a distribuição posterior conjunta (2.3) deve ser recalculada. O recebimento de novas medidas aumenta assim a dimensionalidade dessa distribuição, aumentando a complexidade computacional do cálculo da mesma. Consequentemente, com o passar do tempo, torna-se computacionalmente impossível efetuar o cálculo (ver item 1.3 de [14]). Esse problema pode ser contornado em situações nas quais é possível substituir o cálculo de (2.3) pelo cálculo de distribuições posteriores marginais. É esse o caso da filtragem ótima, pois estamos interessados em calcular a distribuição posterior do estado atual, $p(\mathbf{x}_k|\mathbf{Z}^k)$. Essa distribuição é denominada *distribuição posterior de filtragem*.

2.1 Estimação Sequencial da Distribuição Posterior de Filtragem

O cálculo sequencial da distribuição posterior de filtragem é efetuado em duas etapas, sendo baseado nas seguintes hipóteses:

- A evolução temporal da variável de estado $\mathbf{x}_k$ produz uma cadeia de Markov de primeira ordem;
- A medida $\mathbf{z}_k$ depende somente do estado atual $\mathbf{x}_k$.

Na primeira etapa, denominada *etapa de predição*, a distribuição posterior preditiva, $p(\mathbf{x}_k|\mathbf{Z}^{k-1})$, é obtida por meio do cálculo da distribuição marginal

$$p(\mathbf{x}_k|\mathbf{Z}^{k-1}) = \int p(\mathbf{x}_k, \mathbf{x}_{k-1}|\mathbf{Z}^{k-1})d\mathbf{x}_{k-1} \ , \tag{2.4}$$

onde $\mathbf{Z}^{k-1}$ é o histórico das medidas recebidas até o instante $k-1$, incluindo a medida recebida em $k-1$. Aplicando a regra de Bayes ao integrando, obtemos

$$p(\mathbf{x}_k, \mathbf{x}_{k-1}|\mathbf{Z}^{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Z}^{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}^{k-1}) \ . \tag{2.5}$$

Considerando a hipótese de que o estado obedece a uma cadeia de Markov de primeira ordem, obtemos

$$p(\mathbf{x}_k, \mathbf{x}_{k-1}|\mathbf{Z}^{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}^{k-1}) \ . \tag{2.6}$$

Aplicando a equação de Chapman-Kolmogorov[3], obtemos a distribuição preditiva $p(\mathbf{x}_k|\mathbf{Z}^{k-1})$ como uma relação recursiva entre a probabilidade de transição de estados $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ e a distribuição de probabilidade $p(\mathbf{x}_{k-1}|\mathbf{Z}^{k-1})$ [4]

$$p(\mathbf{x}_k|\mathbf{Z}^{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}^{k-1})d\mathbf{x}_{k-1} \ . \tag{2.7}$$

Na segunda etapa, denominada *etapa de correção*, aplicamos a regra de Bayes à distribuição posterior de filtragem, obtendo

$$p(\mathbf{x}_k|\mathbf{Z}^k) = \frac{p(\mathbf{x}_k, \mathbf{Z}^k)}{p(\mathbf{Z}^k)} \tag{2.8}$$

e

$$p(\mathbf{x}_k|\mathbf{Z}^k) = \frac{p(\mathbf{x}_k, \mathbf{z}_k, \mathbf{Z}^{k-1})}{p(\mathbf{z}_k, \mathbf{Z}^{k-1})} \ . \tag{2.9}$$

A aplicação da regra de Bayes ao numerador e ao denominador da equação anterior resulta em

$$p(\mathbf{x}_k|\mathbf{Z}^k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{Z}^{k-1})p(\mathbf{x}_k|\mathbf{Z}^{k-1})p(\mathbf{Z}^{k-1})}{p(\mathbf{z}_k|\mathbf{Z}^{k-1})p(\mathbf{Z}^{k-1})} \ . \tag{2.10}$$

A simplificação da equação acima produz

$$p(\mathbf{x}_k|\mathbf{Z}^k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{Z}^{k-1})p(\mathbf{x}_k|\mathbf{Z}^{k-1})}{p(\mathbf{z}_k|\mathbf{Z}^{k-1})} \ . \tag{2.11}$$

Considerando a hipótese de que $\mathbf{z}_k$ depende somente de $\mathbf{x}_k$, obtemos

$$\overbrace{p(\mathbf{x}_k|\mathbf{Z}^k)}^{posterior} = \frac{\overbrace{p(\mathbf{z}_k|\mathbf{x}_k)}^{verossimilhanca} \ \overbrace{p(\mathbf{x}_k|\mathbf{Z}^{k-1})}^{anterior}}{\underbrace{p(\mathbf{z}_k|\mathbf{Z}^{k-1})}_{evidencia}} \ , \tag{2.12}$$

que é a relação de atualização recursiva desejada, na qual

$$p(\mathbf{z}_k|\mathbf{Z}^{k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}^{k-1})d\mathbf{x}_k \ . \tag{2.13}$$

Para facilitar a visualização da recursividade, a equação 2.12 pode ser reescrita como

$$\overbrace{p(\mathbf{x}_k|\mathbf{Z}^k)}^{atualizacao} = \overbrace{\mathcal{W}_c(k, k-1)}^{peso} \overbrace{p(\mathbf{x}_k|\mathbf{Z}^{k-1})}^{predicao} \ , \tag{2.14}$$

onde o peso é dado por

$$\mathcal{W}_c(k, k-1) := \frac{p(\mathbf{z}_k|\mathbf{x}_k)}{p(\mathbf{z}_k|\mathbf{Z}^{k-1})} \ . \tag{2.15}$$

As equações 2.7, 2.12 e 2.13 compõem solução recursiva ótima para a filtragem Bayesiana. Por conveniência, repetimos aqui essas equações

[3] $p(\mathbf{x}_k|\mathbf{x}_{k-2}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1})p(\mathbf{x}_{k-2})d\mathbf{x}_{k-1}$

[4] $p(\mathbf{x}_{k-1}|\mathbf{Z}^{k-1})$ é a distribuição posterior no instante $k-1$.

$$
\begin{aligned}
p(\mathbf{x}_k|\mathbf{Z}^{k-1}) &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}^{k-1})d\mathbf{x}_{k-1} \\
p(\mathbf{x}_k|\mathbf{Z}^{k}) &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}^{k-1})}{p(\mathbf{z}_k|\mathbf{Z}^{k-1})} \\
p(\mathbf{z}_k|\mathbf{Z}^{k-1}) &= \int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}^{k-1})d\mathbf{x}_k \ , \qquad (2.16)
\end{aligned}
$$

onde $p(\mathbf{x}_k|\mathbf{Z}^{k-1})$ é a etapa de predição e $p(\mathbf{x}_k|\mathbf{Z}^{k})$ é a etapa de correção.

O problema é que as integrais multidimensionais das equações (2.16) só são tratáveis em situações específicas, como no caso de o sistema dinâmico ser linear e Gaussiano [17]. Neste caso, a solução ótima é obtida pelo Filtro de Kalman (KF). Para sistemas dinâmicos não lineares e/ou não Gaussianos, não há solução recursiva em forma fechada para essas integrais, sendo necessário recorrer a soluções aproximadas.

2.2 Soluções Aproximadas para o Problema de Filtragem Bayesiana

Uma solução aproximada para o problema de filtragem ótima consiste em aproximar as distribuições de filtragem por meio de distribuições Gaussianas. Os filtros que fazem uso dessa estratégia de aproximação são denominados *filtros Gaussianos*. No próximo capítulo, abordaremos a formulação geral dos filtros Gaussianos.

Capítulo 3

Filtros Gaussianos

Conforme visto no Capítulo 2, os filtros Gaussianos permitem obter solução aproximada para o problema de filtragem ótima. Nesse capítulo, voltaremos inicialmente a esse problema, dessa vez com o intuito de justificar o uso do UKF em vez de filtros mais poderosos. Em seguida, apresentaremos a formulação geral do filtro Gaussiano, o que permitirá mostrar que a UT é a peça central do UKF, abrindo caminho para justificar a pesquisa na área de sintonia dos parâmetros dessa transformada.

3.1 Filtragem Ótima

O termo *filtragem ótima* se refere à metodologia usada para estimar o estado de um sistema variante no tempo que é indiretamente observado por meio de um conjunto de medidas ruidosas [18]. Em termos menos rigorosos, mas suficientes para entendimento do conceito, define-se o *estado* de um sistema como sendo o conjunto de variáveis que provê representação completa da condição interna do sistema em um dado instante de tempo [19]. A afirmação de que as medidas são *ruidosas* quer dizer que elas são incertas, no sentido de que elas não são funções determinísticas do estado do sistema.

Para efetuar a filtragem, a evolução temporal do sistema real é modelada como um sistema dinâmico perturbado por um *ruído de processo*. A ideia de existência de um ruído de processo é somente um artifício para lidar com incertezas a respeito da dinâmica do sistema real. Assim, mesmo que a dinâmica do sistema real não seja estocástica, o modelo utilizado pelo filtro incorporará ruído estocástico como mecanismo de lidar com o fato de que não se conhece completamente a dinâmica desse sistema. O mesmo artifício é usado na modelagem das medidas, resultando na existência de um *ruído de medidas*.

A estimação de estados é um problema relevante para a Engenharia, pois encontra aplicações nas mais diversas áreas, incluindo atividades industriais [20], rastreamento de alvos [21], navegação terrestre [22] e atividades aeroespaciais [23]. Parte dos problemas do mundo real pode ser abordada pela teoria da filtragem linear [13], representada pelo onipresente KF. A teoria da filtragem linear é teoricamente bem fundamentada e permite abordar problemas levemente não lineares usando linearização, usualmente por meio do Filtro de Kalman Estendido (EKF) [24][1]. Parte dos problemas relevantes, porém, é de natureza significativamente não linear, aceitando mal o recurso a estratégias baseadas em linearização. A necessidade de lidar com esses problemas levou ao surgimento de algoritmos de filtragem capazes de lidar melhor com a não linearidade, como o Filtro de Partículas (PF) [26] e o UKF.

[1] [25] afirma que o EKF funciona bem em problemas *quasilineares*, que são caracterizados pelo fato de os erros devidos à linearização são pequenos, quando comparados aos erros devidos aos erros de modelamento e ao ruído dos sensores

O PF pode lidar com densidades posteriores de filtragem (2.12) Gaussianas e não Gaussianas. Essa vantagem, porém, tem um preço. O PF é computacionalmente muito mais custoso que o UKF[2]. Assim, se for aceitável aproximar a distribuição posterior de filtragem por uma distribuição Gaussiana, o baixo custo computacional justifica a adoção do UKF.

3.2 Fundamentação Teórica dos Filtros Gaussianos

Filtro Gaussiano [28] é a denominação dada ao filtro que resolve o problema de estimação não linear assumindo que as distribuições em (2.16) podem ser bem aproximadas por distribuições Gaussianas. Bastaria então calcular as médias e covariâncias pertinentes à (2.16) para especificar completamente essas distribuições e, consequentemente, obter solução Gaussiana aproximada para o problema de filtragem Bayesiana não linear.

Mais especificamente, os filtros Gaussianos assumem que a distribuição de filtragem pode ser aproximada por uma distribuição Gaussiana, ou seja,

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \mathcal{N}(\mathbf{x}_k|\mathbf{m}_k, \mathbf{P}_k) \ , \tag{3.1}$$

onde os momentos $\mathbf{m}_k$ e $\mathbf{P}_k$ são calculados por meio das equações do casamento de momentos Gaussiano[3]. Vejamos como são feitos esses cálculos.

Sejam $\mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{P})$ e $\mathbf{q} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ duas variáveis aleatórias Gaussianas quaisquer. Seja então a variável aleatória $\mathbf{y}$, resultante da transformação aditiva

$$\mathbf{y} = \mathbf{g}(\mathbf{x}) + \mathbf{q} \ , \tag{3.2}$$

onde $\mathbf{g}(\cdot)$ é uma função não linear.

De acordo com o casamento de momentos Gaussiano, a distribuição de probabilidade conjunta das variáveis aleatórias $\mathbf{x}$ e $\mathbf{y}$ é dada por

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} \mathbf{m} \\ \boldsymbol{\mu}_M \end{pmatrix} \ \begin{pmatrix} \mathbf{P} & \mathbf{C}_M \\ \mathbf{C}_M^T & \mathbf{S}_M \end{pmatrix}\right] \tag{3.3}$$

onde

$$\begin{aligned} \boldsymbol{\mu}_M &= \int \mathbf{g}(\mathbf{x})\mathcal{N}(\mathbf{x}|\mathbf{m}, \mathbf{P})d\mathbf{x} \\ \mathbf{S}_M &= \int (\mathbf{g}(\mathbf{x}) - \boldsymbol{\mu}_M)(\mathbf{g}(\mathbf{x}) - \boldsymbol{\mu}_M)^T\mathcal{N}(\mathbf{x}|\mathbf{m}, \mathbf{P})d\mathbf{x} + \mathbf{Q} \\ \mathbf{C}_M &= \int (\mathbf{x} - \mathbf{m})(\mathbf{g}(\mathbf{x}) - \boldsymbol{\mu}_M)^T\mathcal{N}(\mathbf{x}|\mathbf{m}, \mathbf{P})d\mathbf{x} \ . \end{aligned} \tag{3.4}$$

3.3 Formulação Geral do Filtro Gaussiano

A formulação geral do filtro Gaussiano combina a aproximação Gaussiana (3.1) e a estrutura do KF [29]. Lembrando que as equações (3.3) e (3.4) assumiram que a variável aleatória $\mathbf{y}$ resulta da transformação aditiva (3.2), vamos contextualizar o problema de fil-

[2]Sob a hipótese de ruídos aditivos, o UKF apresenta complexidade computacional da mesma ordem que a complexidade do EKF [27]

[3]Ver o Capítulo 6 de [14]

tragem Gaussiana, definindo a forma do sistema não linear do qual trataremos desse ponto em diante. Seja então o sistema não linear de tempo discreto

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \ , \end{aligned} \tag{3.5}$$

onde $\mathbf{f}(\cdot)$ e $\mathbf{h}(\cdot)$ são funções conhecidas. O ruído de processo $\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ e o ruído de medida $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ são ruídos brancos, independentes do estado inicial $\mathbf{x}_0$. A formulação geral do filtro Gaussiano para o sistema (3.5) é [28]

Predição

$$\begin{aligned} \mathbf{m}_k^- &= \mathrm{E}[\mathbf{f}(\mathbf{x}_{k-1})] \\ \mathbf{P}_k^- &= \mathrm{E}[(\mathbf{x}_k - \mathbf{m}_k^-)(\mathbf{x}_k - \mathbf{m}_k^-)^T] + \mathbf{Q}_{k-1} \end{aligned} \tag{3.6}$$

Atualização

$$\begin{aligned} \boldsymbol{\mu}_k &= \mathrm{E}[\mathbf{h}(\mathbf{x}_k)] \\ \mathbf{S}_k &= \mathrm{E}[(\mathbf{y}_k - \boldsymbol{\mu}_k)(\mathbf{y}_k - \boldsymbol{\mu}_k)^T] + \mathbf{R}_k \\ \mathbf{C}_k &= \mathrm{E}[(\mathbf{x}_k - \mathbf{m}_k^-)(\mathbf{y}_k - \boldsymbol{\mu}_k)^T] \end{aligned} \tag{3.7}$$

$$\begin{aligned} \mathbf{K}_k &= \mathbf{C}_k \mathbf{S}_k^{-1} \\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k(\mathbf{y}_k - \boldsymbol{\mu}_k) \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T \ . \end{aligned} \tag{3.8}$$

Formulando as expectativas nas equações (3.6) e (3.7) por meio de integrais de momentos (ver Algoritmo 3.21 de [18] e as equações (3.4)), obtemos

Predição

$$\begin{aligned} \mathbf{m}_k^- &= \int \mathbf{f}(\mathbf{x}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1} | \mathbf{m}_{k-1}, \mathbf{P}_{k-1}) d\mathbf{x}_{k-1} \\ \mathbf{P}_k^- &= \int (\mathbf{f}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)(\mathbf{f}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)^T \mathcal{N}(\mathbf{x}_{k-1} | \mathbf{m}_{k-1}, \mathbf{P}_{k-1}) d\mathbf{x}_{k-1} + \mathbf{Q}_{k-1} \end{aligned} \tag{3.9}$$

Atualização

$$\begin{aligned} \boldsymbol{\mu}_k &= \int \mathbf{h}(\mathbf{x}_k) \mathcal{N}(\mathbf{x}_k | \mathbf{m}_k^-, \mathbf{P}_k^-) d\mathbf{x}_k \\ \mathbf{S}_k &= \int (\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)(\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)^T \mathcal{N}(\mathbf{x}_k | \mathbf{m}_k^-, \mathbf{P}_k^-) d\mathbf{x}_k + \mathbf{R}_k \\ \mathbf{C}_k &= \int (\mathbf{x}_k - \mathbf{m}_k^-)(\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)^T \mathcal{N}(\mathbf{x}_k | \mathbf{m}_k^-, \mathbf{P}_k^-) d\mathbf{x}_k \end{aligned} \tag{3.10}$$

$$\begin{aligned}
\mathbf{K}_k &= \mathbf{C}_k \mathbf{S}_k^{-1} \\
\mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k(\mathbf{y}_k - \boldsymbol{\mu}_k) \\
\mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T
\end{aligned} \tag{3.11}$$

3.4 Alguns Filtros Gaussianos

É interessante notar que as integrais (3.9) e (3.10) têm a forma geral

$$\int \mathbf{g}(\mathbf{x}) \mathcal{N}(\mathbf{x}|\mathbf{m}, \mathbf{P}) d\mathbf{x} \ , \tag{3.12}$$

onde $\mathbf{g}(\cdot)$ é uma função não linear conhecida. A formulação (3.12) permite o uso de diferentes métodos para a solução das integrais (3.9) e (3.10). Diferentes métodos de solução dão origem a diferentes filtros Gaussianos ([28], [29]).

Por exemplo, as regras de cubatura dão origem ao Filtro de Kalman de Cubatura (CKF) e a regra de quadratura de Gauss-Hermite dá origem ao Filtro de Kalman de Gauss-Hermite (GHKF) [28]. Por sua vez, o EKF recorre à linearização por expansão em série de Taylor de primeira ordem das equações do sistema (3.5), enquanto o UKF utiliza a UT. Assim, tanto o EKF como o UKF se enquadram na moldura geral dos filtros Gaussianos.

Capítulo 4

O Filtro de Kalman *Unscented*

Vimos no Capítulo 3 que tanto o EKF como o UKF são filtros Gaussianos e que a diferença entre eles reside na maneira com que implementam a aproximação Gaussiana para as distribuições em (2.16).

No EKF, a aproximação Gaussiana é obtida por meio de aproximações das não linearidades pelas respectivas expansões em série de Taylor de primeira ordem. Para entender melhor o funcionamento do EKF e suas limitações, vejamos como a expansão em série linear de Taylor é usada para calcular a propagação, pela função não linear $\mathbf{g}(\cdot)$, dos dois primeiros momentos de uma variável aleatória $\mathbf{x} \in \Re^{n_x}$, distribuída segundo $\mathbf{x} = \mathcal{N}(\boldsymbol{\mu}_\mathbf{x}, \mathbf{P}_\mathbf{x})$. Assim, dados os dois primeiros momentos de $\mathbf{x}$, usaremos a expansão em série de Taylor da função $\mathbf{g}(\cdot)$ para calcular os dois primeiros momentos da variável transformada $\mathbf{y} = \mathbf{g}(\mathbf{x})$.

Se fizermos $\mathbf{x} = \boldsymbol{\mu}_\mathbf{x} + \boldsymbol{\delta}_\mathbf{x}$, sendo $\boldsymbol{\delta}_\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_\mathbf{x})$, a expansão de $\mathbf{g}(\cdot)$ em série de Taylor de primeira ordem é dada por[1]

$$\mathbf{g}(\mathbf{x}) = \mathbf{g}(\boldsymbol{\mu}_\mathbf{x} + \boldsymbol{\delta}_\mathbf{x}) \approx g(\boldsymbol{\mu}_\mathbf{x}) + \mathbf{G}_\mathbf{x}(\boldsymbol{\mu}_\mathbf{x})\boldsymbol{\delta}_\mathbf{x} \ , \tag{4.1}$$

onde

$$\mathbf{G}_\mathbf{x}(\boldsymbol{\mu}_\mathbf{x}) = \left.\frac{d\mathbf{g}(\mathbf{x})}{d\mathbf{x}}\right|_{\mathbf{x}=\boldsymbol{\mu}_\mathbf{x}} \tag{4.2}$$

é a matriz Jacobiana de $\mathbf{g}(\mathbf{x})$. A aproximação linear $\hat{\mathbf{g}}(\mathbf{x})$ para $\mathbf{g}(\mathbf{x})$ é então dada por

$$\hat{\mathbf{g}}(\mathbf{x}) = \mathbf{g}(\boldsymbol{\mu}_\mathbf{x}) + \mathbf{G}_\mathbf{x}(\boldsymbol{\mu}_\mathbf{x})\boldsymbol{\delta}_\mathbf{x} \ . \tag{4.3}$$

O primeiro momento da variável $\mathbf{y} = \mathbf{g}(\mathbf{x})$ é aproximado pelo primeiro momento da expansão (4.3)

$$\begin{aligned} \mathrm{E}[\hat{\mathbf{g}}(\mathbf{x})] &= \mathrm{E}[\mathbf{g}(\boldsymbol{\mu}_\mathbf{x}) + \mathbf{G}_\mathbf{x}(\boldsymbol{\mu}_\mathbf{x})\boldsymbol{\delta}_\mathbf{x}] \\ &= \mathbf{g}(\boldsymbol{\mu}_\mathbf{x}) \ , \end{aligned} \tag{4.4}$$

enquanto o segundo momento central é aproximado por

$$\mathrm{E}\left[(\hat{\mathbf{g}}(\mathbf{x}) - \mathrm{E}[\hat{\mathbf{g}}(\mathbf{x})])(\hat{\mathbf{g}}(\mathbf{x}) - \mathrm{E}[\hat{\mathbf{g}}(\mathbf{x})])^T\right] = \mathrm{E}\left[(\hat{\mathbf{g}}(\mathbf{x}) - \mathbf{g}(\boldsymbol{\mu}_\mathbf{x}))(\hat{\mathbf{g}}(\mathbf{x}) - \mathbf{g}(\boldsymbol{\mu}_\mathbf{x}))^T\right] \tag{4.5}$$

$$= \mathbf{G}_\mathbf{x}(\boldsymbol{\mu}_\mathbf{x})\mathbf{P}_\mathbf{x}\mathbf{G}_\mathbf{x}^T(\boldsymbol{\mu}_\mathbf{x}) \ . \tag{4.6}$$

Quando o sistema dinâmico (3.5) é pronunciadamente não linear, as aproximações (4.3)

[1] Ver item 5.1 de [14]

para as funções de transição de estados e de medida não são suficientemente precisas, prejudicando o cálculo dos momentos aproximados dos vetores de estado de medida e, consequentemente, impedindo o bom funcionamento do filtro.

O UKF [4] é capaz de lidar melhor com sistemas não lineares. Em vez de construir aproximações lineares para as funções não lineares do sistema dinâmico (3.5), como faz o EKF, o UKF utiliza a UT para aproximar diretamente os dois primeiros momentos da distribuição alvo, o que difere significativamente da estratégia de linearizar $\mathbf{g}(\cdot)$ por expansão em série de Taylor de primeira ordem. Para entender o funcionamento da UT, vamos retomar o problema de estimar os dois primeiros momentos de

$$\mathbf{y} = \mathbf{g}(\mathbf{x}) \ , \tag{4.7}$$

onde $\mathbf{x} = \mathcal{N}(\boldsymbol{\mu}_\mathbf{x}, \mathbf{P}_\mathbf{x})$.

A UT escolhe deterministicamente pontos no espaço de estados do sistema dinâmico que capturam a média e a covariância da distribuição de $\mathbf{x}$. Esses pontos são denominados *pontos sigma*. Os pontos sigma são então propagados pela não linearidade $\mathbf{g}(\cdot)$, e a média e a covariância de $\mathbf{y}$ são estimadas a partir dos pontos sigma propagados. Detalhemos agora o funcionamento da UT.

4.1 A Transformada *Unscented* Escalada

A UT gera os $2n_x + 1$ vetores $\mathcal{X}_i$ denominados *pontos sigma*

$$\begin{aligned} \mathcal{X}_0 &= \boldsymbol{\mu}_\mathbf{x} \\ \mathcal{X}_i &= \boldsymbol{\mu}_\mathbf{x} + (\sqrt{(n_x+\lambda)\mathbf{P}_\mathbf{x}})_i \quad i \in 1 \ldots n_x \\ \mathcal{X}_{i+n_x} &= \boldsymbol{\mu}_\mathbf{x} - (\sqrt{(n_x+\lambda)\mathbf{P}_\mathbf{x}})_i \quad i \in 1 \ldots n_x \end{aligned} \tag{4.8}$$

onde $(\sqrt{(n_x+\lambda)\mathbf{P}_\mathbf{x}})_i$ é a i-ésima coluna da matriz $\sqrt{(n_x+\lambda)\mathbf{P}_\mathbf{x}}$, tal que

$$(\sqrt{(n_x+\lambda)\mathbf{P}_\mathbf{x}})(\sqrt{(n_x+\lambda)\mathbf{P}_\mathbf{x}})^T = (n_x+\lambda)\mathbf{P}_\mathbf{x} \ . \tag{4.9}$$

Em seguida, a transformação não linear (4.7) é aplicada aos pontos (4.8), produzindo os pontos sigma transformados $\mathcal{Y}_i = \mathbf{g}(\mathcal{X}_i)$. Os pontos $\mathcal{Y}_i$ são então utilizados para estimar a média $\boldsymbol{\mu}_\mathbf{y}$ e a covariância $\mathbf{P}_\mathbf{y}$ de $\mathbf{y}$, bem como a covariância cruzada $\mathbf{P}_\mathbf{xy}$ de $\mathbf{x}$ e $\mathbf{y}$

$$\boldsymbol{\mu}_\mathbf{y} = w_0^{(m)}\mathcal{Y}_0 + \sum_{i=1}^{2n_x} w_i^{(m)}\mathcal{Y}_i \tag{4.10}$$

$$\mathbf{P}_\mathbf{y} = w_0^{(c)}(\mathcal{Y}_0 - \boldsymbol{\mu}_\mathbf{y})(\mathcal{Y}_0 - \boldsymbol{\mu}_\mathbf{y})^T + \sum_{i=1}^{2n_x} w_i^{(c)}(\mathcal{Y}_i - \boldsymbol{\mu}_\mathbf{y})(\mathcal{Y}_i - \boldsymbol{\mu}_\mathbf{y})^T \tag{4.11}$$

$$\mathbf{P}_\mathbf{xy} = w_0^{(c)}(\mathcal{X}_0 - \boldsymbol{\mu}_\mathbf{x})(\mathcal{Y}_0 - \boldsymbol{\mu}_\mathbf{y})^T + \sum_{i=1}^{2n_x} w_i^{(c)}(\mathcal{X}_i - \boldsymbol{\mu}_\mathbf{x})(\mathcal{Y}_i - \boldsymbol{\mu}_\mathbf{y})^T \tag{4.12}$$

$$
\begin{aligned}
w_0^{(m)} &= \frac{\lambda}{n_x+\lambda} \\
w_0^{(c)} &= \frac{\lambda}{n_x+\lambda} + (1-\alpha^2+\beta) \\
w_i^{(m)} &= \frac{1}{2(n_x+\lambda)}\ , \quad i \in 1\ldots 2n_x \\
w_i^{(c)} &= \frac{1}{2(n_x+\lambda)}\ , \quad i \in 1\ldots 2n_x
\end{aligned} \tag{4.13}
$$

onde $\lambda = \alpha^2(n_x+\kappa) - n_x$ é um parâmetro de escalamento, α define o espalhamento dos pontos sigma em torno de $\boldsymbol{\mu}_{\mathbf{x}}$, κ é um parâmetro de escalamento secundário e β, segundo [27], pode ser usado para incorporar conhecimento a priori sobre a distribuição de $\mathbf{x}$.

4.2 O Filtro de Kalman *Unscented*

A UT é o elemento central do UKF. Antes de apresentar o algoritmo do UKF, ressaltemos que, nas variáveis relativas aos estados e às medidas, o sobrescrito indica o instante de tempo ao qual a variável se refere e os subscritos $\mathbf{x}$ e $\mathbf{y}$ indicam se a variável se refere respectivamente ao estado ou à medida. Assim, $\boldsymbol{\mu}_{\mathbf{x}}^k$ se refere à média de $\mathbf{x}_k$, o vetor de estado, no instante k. Considerando então o sistema dinâmico aditivo de tempo discreto (3.5), estes são os passos executados pela forma aditiva do UKF [2] a cada instante de tempo k.

Predição

Passo 1 Usando as equações (4.8), calcule os pontos sigma $\mathcal{X}_i$. Por conveniência, as equações (4.8) são repetidas aqui

$$
\begin{aligned}
\mathcal{X}_0^{k-1} &= \boldsymbol{\mu}_{\mathbf{x}}^{k-1} \\
\mathcal{X}_i^{k-1} &= \boldsymbol{\mu}_{\mathbf{x}}^{k-1} + (\sqrt{(n_x+\lambda)\mathbf{P}_{\mathbf{x}}^{k-1}})_i \quad i \in 1\ldots n_x \\
\mathcal{X}_{i+n_x}^{k-1} &= \boldsymbol{\mu}_{\mathbf{x}}^{k-1} - (\sqrt{(n_x+\lambda)\mathbf{P}_{\mathbf{x}}^{k-1}})_i \quad i \in 1\ldots n_x
\end{aligned} \tag{4.14}
$$

Passo 2 Propague os pontos sigma pelo modelo dinâmico, originando

$$
\hat{\mathcal{X}}_i^k = \mathbf{f}(\mathcal{X}_i^{k-1})\ ,\ i = 0,\ldots,2n_x \tag{4.15}
$$

Passo 3 Computar a média $\hat{\boldsymbol{\mu}}_{\mathbf{x}}^k$ e a covariância $\hat{\mathbf{P}}_{\mathbf{x}}^k$ preditas para o estado atual do sistema

$$
\hat{\boldsymbol{\mu}}_{\mathbf{x}}^k = \sum_{i=0}^{2n_x} w_i^{(m)} \hat{\mathcal{X}}_i^k \tag{4.16}
$$

$$
\hat{\mathbf{P}}_{\mathbf{x}}^k = \sum_{i=0}^{2n_x} w_i^{(c)} (\hat{\mathcal{X}}_i^k - \hat{\boldsymbol{\mu}}_{\mathbf{x}}^k)(\hat{\mathcal{X}}_i^k - \hat{\boldsymbol{\mu}}_{\mathbf{x}}^k)^T + \mathbf{Q}_{k-1}\ , \tag{4.17}
$$

[2] Ver seção 5.6 de [14]

sendo os pesos $w_i^{(m)}$ e $w_i^{(c)}$ dados por (4.13).

Atualização

Passo 1 Utilizando a média (4.16) e a covariância (4.17), calcule novos pontos sigma de acordo com

$$\begin{aligned}
\mathcal{X}_0^k &= \hat{\boldsymbol{\mu}}_{\mathbf{x}}^k \\
\mathcal{X}_i^k &= \hat{\boldsymbol{\mu}}_{\mathbf{x}}^k + (\sqrt{(n_x+\lambda)\hat{\mathbf{P}}_{\mathbf{x}}^k})_i \quad i \in 1 \ldots n_x \\
\mathcal{X}_{(i+n_x)}^k &= \hat{\boldsymbol{\mu}}_{\mathbf{x}}^k - (\sqrt{(n_x+\lambda)\hat{\mathbf{P}}_{\mathbf{x}}^k})_i \quad i \in 1 \ldots n_x \ .
\end{aligned} \tag{4.18}$$

Passo 2 Propague os pontos sigma pelo modelo de medidas, originando

$$\hat{\mathcal{Y}}_i^k = \mathbf{h}(\mathcal{X}_i^k) \ , \ i = 0, \ldots, 2n_x \ . \tag{4.19}$$

Passo 3 Computar a média $\hat{\boldsymbol{\mu}}_y^k$ e a covariância $\hat{\mathbf{P}}_y^k$ preditas para a medida a ser recebida. Calcular também a covariância cruzada entre as predições do estado e da medida

$$\hat{\boldsymbol{\mu}}_{\mathbf{y}}^k = \sum_{i=0}^{2n_x} w_i^{(m)} \hat{\mathcal{Y}}_i^k \tag{4.20}$$

$$\hat{\mathbf{P}}_{\mathbf{y}}^k = \sum_{i=0}^{2n_x} w_i^{(c)} (\hat{\mathcal{Y}}_i^k - \hat{\boldsymbol{\mu}}_{\mathbf{y}}^k)(\hat{\mathcal{Y}}_i^k - \hat{\boldsymbol{\mu}}_{\mathbf{y}}^k)^T + \mathbf{R}_k \tag{4.21}$$

$$\hat{\mathbf{C}}_{\mathbf{y}}^k = \sum_{i=0}^{2n_x} w_i^{(c)} (\mathcal{X}_i^k - \hat{\boldsymbol{\mu}}_{\mathbf{x}}^k)(\hat{\mathcal{Y}}_i^k - \hat{\boldsymbol{\mu}}_{\mathbf{y}}^k)^T \ , \tag{4.22}$$

sendo os pesos $w_i^{(m)}$ e $w_i^{(c)}$ dados por (4.13).

Passo 4 Computar o ganho $\mathbf{K}_k$, a média $\boldsymbol{\mu}_{\mathbf{x}}^k$ e a covariância $\mathbf{P}_{\mathbf{x}}^k$ finais do estado filtrado, condicionadas à medida recebida $\mathbf{y}_k$

$$\mathbf{K}_k = \hat{\mathbf{C}}_{\mathbf{y}}^k (\hat{\mathbf{P}}_{\mathbf{y}}^k)^{-1} \tag{4.23}$$

$$\boldsymbol{\mu}_{\mathbf{x}}^k = \hat{\boldsymbol{\mu}}_{\mathbf{x}}^k + \mathbf{K}_k[\mathbf{y}_k - \hat{\boldsymbol{\mu}}_y^k] \tag{4.24}$$

$$\mathbf{P}_{\mathbf{x}}^k = \hat{\mathbf{P}}_{\mathbf{x}}^k - \mathbf{K}_k \hat{\mathbf{P}}_{\mathbf{y}}^k \mathbf{K}_k^T \tag{4.25}$$

4.3 Por Que Sintonizar o UKF ?

O algoritmo do UKF deixa claro que o funcionamento do filtro está intimamente ligado aos valores atribuídos aos parâmetros α, β e κ [30]. Na falta de diretrizes teóricas que ajudem na seleção de valores adequados para esses parâmetros, a tarefa de sintonia se torna bastante difícil. Essas condições fazem da sintonia automática do UKF área ativa de pesquisa, como veremos no Capítulo 5.

As matrizes de covariância dos ruídos de processo e de medidas também influenciam definitivamente no funcionamento do UKF. Em razão disso, podem ser incluídas no conjunto de parâmetros a serem sintonizados, como descrito no Capítulo 10.

Capítulo 5

Revisão Bibliográfica

Neste capítulo, são descritas técnicas de ajuste dos parâmetros do UKF encontradas na literatura recente, incluindo algumas técnicas de adaptação do ruído de processo desenvolvidas especificamente no contexto do KF/EKF. A lista de trabalhos não é exaustiva, pois a sintonia do UKF é área ativa de pesquisa. Listamos aqui os trabalhos publicados recentemente que consideramos mais relevantes. Antes de iniciar a revisão propriamente dita, cabe resumir as recomendações usuais para a sintonia dos parâmetros da UT, α, β e κ [31]:

1. Escolha $\kappa \geq 0$, de modo a guarantir que a matriz de covariância seja positiva definida. O valor específico de κ não é crucial, sendo recomendada a escolha $\kappa = 0$;

2. Escolha $0 \leq \alpha \leq 1$. Como α controla o espalhamento dos pontos sigma, seu valor deve ser pequeno, de modo a evitar amostrar efeitos não locais que ocorrem quando o sistema é pronunciadamente não linear;

3. O parâmetro β é um peso não negativo, o qual pode ser usado para incorporar conhecimento existente a respeito dos momentos de ordem mais alta da distribuição do estado. Para estados cuja distribuição anterior é Gaussiana, deve-se fazer $\beta = 2$. Esse parâmetro pode ser utilizado para controlar o peso da cauda da distribuição posterior, pois controla o erro na curtose da distribuição.

5.1 Aprendizado *offline* discriminativo das covariâncias dos ruídos de processo e de medida

[32] propõe o uso de algoritmos de aprendizado discriminativo[1]([33], [34]) para ajustar automaticamente as covariâncias dos ruídos de processo $\mathbf{Q}$ e de medida $\mathbf{R}$ de um KF/EKF. A pesquisa é justificada, dentre outras razões, pelo argumento de que os termos de ruído podem ser usados para capturar os efeitos causados por perturbações não modeladas no sistema dinâmico. Dentre essas perturbações, destacam-se eventuais deficiências no modelo matemático do sistema dinâmico, erros resultantes da discretização do tempo e efeitos da linearização adotada no EKF.

Os vários algoritmos propostos adotam como estratégia comum encontrar as matrizes de covariância de ruído que maximizam a precisão das estimativas produzidas pelo filtro. Todos, exceto um dos algoritmos, demandam a disponibilidade de medidas de alta precisão do estado ou de parte dele durante a fase de treinamento. O único algoritmo que não demanda a

[1] Treinamento discriminativo é uma estratégia que foca diretamente na maximização do critério de desempenho de interesse.

existência desses dados de alta qualidade maximiza a verossimilhança das medidas recebidas $\mathbf{y}_{0:T}$ condicionada à sequência $\mathbf{u}_{1:t}$ de controles aplicados ao robô

$$[\mathbf{Q}_{opt}, \mathbf{R}_{opt}] = \underset{R,Q}{arg\,max}\ log(\mathbf{p}(\mathbf{y_{0:T}}|\mathbf{u_{1:t}}))\ . \tag{5.1}$$

Os autores ressaltam que, por não envolver o uso de medidas de alta precisão do estado do sistema durante a fase de treinamento, a maximização da verossimilhança condicional das medidas não garante que as estimativas de estado produzidas pelo filtro sejam precisas, que é o que realmente importa.

Apesar disso, essa abordagem é importante para o nosso trabalho, pois trata a sintonia dos parâmetros que governam o filtro como um problema de maximização da verossimilhança condicional das medidas. A maximização da verossimilhança das medidas como critério de otimização dos parâmetros do filtro foi usado em várias abordagens de sintonia propostas posteriormente([35], [1], [3], [36]).

5.2 Aprendizado *offline* discriminativo de todos os parâmetros do UKF

[30] propõem uma técnica de aprendizado *offline* para obtenção dos parâmetros ótimos de um UKF, com foco na localização precisa de robôs móveis. Além de ajustar as covariâncias dos ruídos de processo e de medida, de vital importância em problemas de filtragem em robótica móvel, a técnica proposta ajusta também os parâmetros α, β e κ da UT.

A abordagem proposta é baseada nas abordagens discriminativas investigadas em [32] e demanda a existência de medidas acuradas do estado do sistema para avaliação do desempenho do filtro. Segundo os autores, os resultados obtidos por eles são consistentemente melhores que os obtidos por ajustes manuais e pelas abordagens que ajustam somente as covariâncias de ruído e de medida propostas por [32].

A técnica proposta usa medidas precisas $\mathbf{y}_t$ do vetor de estado $\mathbf{x}_t$ ou de parte dele. Os parâmetros ótimos são obtidos por meio da maximização da verossimilhança

$$[\mathbf{R_{op}}, \mathbf{Q_{op}}, \mathbf{\Omega_{op}}] = \underset{R,Q,\mathbf{\Omega}}{arg\,max} \sum_{t=0}^{T} log(\mathbf{p}(\mathbf{y_t}|\mathbf{z_{0:t}}, \mathbf{u_{1:t}})) \tag{5.2}$$

onde $\mathbf{\Omega}_{op}$ representa o conjunto de valores ótimos para os parâmetros da UT, $\mathbf{z}_{0:T}$ representa a sequência de observações ruidosas que estarão normalmente disponíveis para o filtro e $\mathbf{u}_{1:T}$ representa a sequência de controles aplicados ao robô.

A abordagem proposta em [30] representa avanço em relação ao trabalho de [32], pois sintoniza todos os parâmetros do filtro. O ponto negativo é a dependência em relação a medidas de alta precisão do estado do sistema para o processo de otimização.

5.3 Ajuste *online* de κ por maximização da verossimilhança condicional da medida

Os autores de [1] criticam as abordagens baseadas no aprendizado discriminativo propostas em [30]. Segundo esses autores, o ponto positivo da abordagem é que, por ser *offline*, o processo de aprendizado não impacta na velocidade de execução do filtro. O ponto fraco da abordagem seria a necessidade de haver medidas muito precisas do estado ou de parte

dele, as quais são essenciais para o processo de treinamento. Além disso, essa metodologia implicaria em que mudanças no ponto de operação do sistema dinâmico demandariam a realização de nova otimização, fazendo com que o método seja de pouca valia em problemas nos quais o ponto de operação do sistema não linear varia.

Assim, [1] propõe escolher, a cada medida recebida pelo UKF, o valor do parâmetro de escalamento κ que maximiza o critério de otimização. O valor de κ é escolhido de um conjunto discreto $\mathbf{K} = \{\kappa_1, \ldots, \kappa_{n_K}\}$ de valores considerados factíveis. Valores factíveis de κ são aqueles que garantem que a matriz de covariância estimada pelo filtro seja positiva definida. Vale ressaltar que [1] trabalha com a versão simplificada da UT[2] cujo único parâmetro é κ[4].

Sendo $\mathbf{y}_k$ a medida ruidosa recebida pelo UKF no instante k, os autores usam como critério de otimização a verossimilhança dessa medida, a qual é aproximada pela distribuição Gaussiana

$$\hat{p}(\mathbf{y_k}|\kappa_i) \simeq \mathcal{N}(\mathbf{y}_k|\hat{\mathbf{y}}_k(\kappa_i), \mathbf{P}_k(\kappa_i)) \ , \tag{5.3}$$

onde $\hat{\mathbf{y}}_k(\kappa_i)$ e $\mathbf{P}_k(\kappa_i)$ são respectivamente a medida predita pelo UKF e a covariância dessa previsão, quando o parâmetro de escalamento da UT é $\kappa = \kappa_i$, sendo $\kappa_i \in \mathbf{K}$.

5.4 Ajuste *online* de κ por minimização do erro de predição de medida

Em [37], os autores ressaltam que a sintonia de κ baseada na maximização da verossimilhança (5.3) apresenta uma fraqueza relevante: a hipótese de que é possível aproximar a verossimilhança por uma densidade Gaussiana passível de ser calculada analiticamente. Assim, os autores estendem o trabalho de [1], propondo o uso de dois critérios baseados somente em momentos preditivos.

O primeiro critério consiste na minimização do erro de predição de medida $\tilde{\mathbf{y}}_k(\kappa_i) = \mathbf{y}_k - \hat{\mathbf{y}}_k(\kappa_i)$ e é dado por

$$\kappa_k = \underset{\kappa_i \in K}{arg\,min} \left(\tilde{\mathbf{y}}_k(\kappa_i)\tilde{\mathbf{y}}_k(\kappa_i)^T\right) \ . \tag{5.4}$$

O segundo critério consiste em incluir a minimização da covariância da predição de medida $\mathbf{P}_k(\kappa)$, sendo dado por

$$\kappa_k = \underset{\kappa_i \in K}{arg\,min} \left(\tilde{\mathbf{y}}_k(\kappa_i)\tilde{\mathbf{y}}_k(\kappa_i)^T + \mathrm{trace}\mathbf{P}_k(\kappa_i)\right) \tag{5.5}$$

A estratégia de adaptação online do parâmetro κ é melhorada em [3], com a proposição de mecanismos de adaptação menos custosos em termos computacionais, mas, em essência, a estratégia permanece a mesma.

5.5 Sintonia dos Parâmetros da UT Usando Otimização por Modelos Substitutos

A ideia de abordar a sintonia dos parâmetros do UKF como um problema de otimização baseada em modelos substitutos foi apresentada em [38] e em [36]. A estratégia consiste

[2] A versão simplificada da UT é obtida fazendo $\alpha = 1$ e $\beta = 0$.

em usar técnicas de otimização baseada em modelos para otimizar uma função objetivo que mapeia um ponto no espaço de parâmetros da UT à qualidade das estimativas produzidas pelo UKF quando sintonizado com esse ponto.

Calcular a função objetivo em um dado ponto do espaço de parâmetros da UT consiste em executar um UKF sintonizado com os valores definidos por esse ponto. Então, com base nas medidas produzidas pelo filtro, calcular a verossimilhança de uma série temporal formada por medidas ruidosas obtidas do sistema dinâmico.

As contribuições de ([38], [36]) consistiram em conceber uma função objetivo que permitisse utilizar as ferramentas da área de otimização baseada em modelos para a sintonia da UT e propor a utilização de um algoritmo baseado em gradientes para essa otimização.

A abordagem proposta por ([38], [36]) adota elementos importantes utilizados em outros trabalhos. Notadamente, a abordagem consiste na sintonia *offline* dos parâmetros do filtro por um processo de otimização que usa verossimilhança das medidas como função objetivo, como inicialmente proposto em [32] e depois estendido em [30].

Capítulo 6

Abordando a Sintonia do UKF Como Um Problema de Otimização

O principal objetivo deste trabalho é melhorar a qualidade das estimativas de estado produzidas pelo UKF, sem impor carga computacional extra ao filtro. Desse modo, o UKF preservaria um de seus principais atrativos, que é o baixo custo computacional. Uma maneira de atingir esse objetivo é abordar a sintonia dos parâmetros do UKF como um problema de otimização *offline*.

Para otimizar os parâmetros do UKF, seria então necessário definir uma função objetivo escalar $g(\cdot)$ que avaliasse adequadamente um dado ponto do espaço de parâmetros do filtro. A avaliação atribuída ao ponto teria de ser indicativa da qualidade das estimativas de estado produzidas pelo UKF, quando sintonizado com o ponto em questão. O processo de sintonia consistiria assim em encontrar o ponto do espaço de parâmetros do UKF que maximiza a função objetivo $g(\cdot)$.

6.1 A Função Objetivo

Os algoritmos de sintonia propostos neste trabalho adotam a mesma função objetivo usada em [38], com a diferença de que, aqui, o espaço de parâmetros da função objetivo poderá incluir os elementos diagonais das matrizes de covariância dos ruídos de processo e de medida, conforme será visto no Capítulo 10. Assim, falaremos em geral de espaço de parâmetros do UKF, em vez de espaço de parâmetros da UT.

Para um dado ponto $\boldsymbol{\theta} \in \Re^{n_\theta}$ no espaço de parâmetros do UKF, essa função objetivo mede a verossimilhança de que uma sequência temporal de medidas obtidas do sistema dinâmico seja predita pelo UKF sintonizado com $\boldsymbol{\theta}$. A mesma estratégia foi aplicada na sintonia *offline* das matrizes de covariância dos ruídos do EKF em [32] e no ajuste *online* do parâmetro κ da UT em ([35],[1],[37],[2],[3]).

Mais detalhadamente, para um dado ponto $\boldsymbol{\theta}$, essa função objetivo executa o UKF sintonizado com $\boldsymbol{\theta}$ em uma ou mais séries temporais

$$\mathbf{Y}_i = \{\mathbf{y}_i(1), \mathbf{y}_i(2), \ldots, \mathbf{y}_i(T)\}\,,\ i = 1, \ldots, n_Y \tag{6.1}$$

onde $\mathbf{y}_i(t)$, $t = 1, \ldots, T$, são medidas ruidosas sequenciais do estado do sistema dinâmico. Para cada série temporal $\mathbf{Y}_i$, as médias e covariâncias das medidas preditas pelo filtro sintonizado com $\boldsymbol{\theta}$ são usadas para calcular o logaritmo da verossimilhança condicional $p(\mathbf{Y}_i|\boldsymbol{\theta})$, dada por

$$\log p(\mathbf{Y}_i|\boldsymbol{\theta}) = \frac{\sum_{t=1}^{T} \log\left(\mathcal{N}(\mathbf{y}_i(t)|\hat{\mathbf{y}}_i(t,\boldsymbol{\theta}), \mathbf{S}_i(t,\boldsymbol{\theta}))\right)}{T}, \tag{6.2}$$

onde $\hat{\mathbf{y}}_i(t,\boldsymbol{\theta})$ e $\mathbf{S}_i(t,\boldsymbol{\theta})$ são respectivamente a média e a covariância da medida predita pelo UKF sintonizado com $\boldsymbol{\theta}$ no instante discreto t.

A sequência de cálculos executados por essa função objetivo é detalhada no Algoritmo 1. Os dados de entrada são os parâmetros $\boldsymbol{\theta}$ do UKF, as séries temporais $\mathbf{Y} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_{n_Y}\}$ que servem de dados de treinamento, e tanto a média $\mathbf{M}_0$ como a covariância $\mathbf{P}_0$ do estado inicial.

Algorithm 1 Função objetivo $g(\cdot)$

1: **procedure** GOALFUN($\boldsymbol{\Theta}, \mathbf{Y}, \mathbf{M}_0, \mathbf{P}_0$)
2: **for** each $\mathbf{Y}_i \in \mathbf{Y}$ **do**
3: $\left[\hat{\mathbf{Y}}_i, \mathbf{S}_i\right] \leftarrow$ UKF($\boldsymbol{\theta}, \mathbf{Y}_i, \mathbf{M}_0, \mathbf{P}_0$) ▷ Executa o UKF sintonizado com $\boldsymbol{\Theta}$
4: loglik($\mathbf{Y}_i$) $= \log p(\mathbf{Y}_i, \boldsymbol{\theta})$ ▷ Calcula o logaritmo da verossimilhança condicional da série temporal $\mathbf{Y}_i$, que é dado por (6.2)
5: $\mathrm{w}_{\boldsymbol{\theta}} \leftarrow \sum_1^{n_Y}$ loglik($\mathbf{Y}_i$) ▷ Calcula o somatório das verossimilhanças das n_Y séries de treinamento $\mathbf{Y}_i$
6: **return** $\mathrm{w}_{\boldsymbol{\theta}}$ ▷ Retorna o somatório calculado.

A função objetivo descrita no Algoritmo 1 depende do cálculo da verossimilhança feito na equação (6.2). A fim de tornar o processo de sintonia o mais genérico possível, trataremos a função objetivo como uma função caixa preta. Desse modo, assumiremos que não é possível obter expressões analíticas para as derivadas dessa função.

6.2 Algoritmo Geral do Processo de Sintonia do UKF

Definida a função objetivo, o processo de sintonia do UKF consiste em utilizar um algoritmo de otimização para, recursivamente, definir novos $\boldsymbol{\theta}$ nos quais avaliar $g(\cdot)$, com o intuito de encontrar o ponto $\boldsymbol{\theta}^\star$ que maximiza $g(\cdot)$. A moldura geral do processo de sintonia recursiva do UKF é descrita no Algoritmo 2. Como em qualquer processo de otimização, é necessário que o usuário defina alguns parâmetros. Vejamos como isso pode ser feito:

budget - Em princípio, quanto maior o número de avaliações da função objetivo, maiores as chances de se encontrar uma boa sintonia.

range - Quanto maior a faixa de valores permitidos para os parâmetros do UKF, maiores as chances de incluir no espaço de parâmetros pontos que produziriam boas sintonias. Em contrapartida, mais difícil seria encontrar esses pontos.

Y - As séries temporais de treinamento devem ser escolhidas de modo a representar da melhor maneira possível as situações que o filtro pode encontrar. Esse tema é abordado na próxima seção.

Θ - Em princípio, quanto mais amostras iniciais, melhor. O problema é que em geral a quantidade de amostras iniciais é limitada por fatores diversos. Em razão disso, a definição das amostras iniciais é uma área de pesquisa (ver Capítulo 1 de [39]).

Algorithm 2 Sintonia do UKF

1: **procedure** SINTONIAUKF
2: *budget* ← Define o número máximo de avaliações da função objetivo
3: *range* ← Define os limites permitidos para os valores que os parâmetros do UKF podem assumir
4: $\mathbf{Y} \leftarrow \{\mathbf{Y}_1, \ldots, \mathbf{Y}_{n_Y}\}$. ▷ Obtém as séries de treinamento
5: $\boldsymbol{\Theta} \leftarrow \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$ ▷ Seleciona um conjunto inicial de pontos no espaço de parâmetros do UKF
6: $\mathbf{W}_\Theta \leftarrow g(\boldsymbol{\Theta})$ ▷ Avalia a função objetivo nos pontos escolhidos
7: **while** $j \leftarrow 1, budget$ **do**
8: $\boldsymbol{\theta}_j \leftarrow \text{search}(\Theta, \mathrm{W}_\Theta, \text{range})$ ▷ Efetua busca pelo próximo ponto onde avaliar a função objetivo $g(\cdot)$
9: $\mathbf{w}_j \leftarrow g(\boldsymbol{\theta}_j)$ ▷ Avalia a função objetivo no ponto escolhido
10: $\boldsymbol{\Theta} \leftarrow \{\boldsymbol{\Theta}; \boldsymbol{\theta}_j\}$ ▷ Atualiza o conjunto de pontos nos quais a função objetivo foi avaliada
11: $\mathbf{W}_\Theta \leftarrow \{\mathbf{W}_\Theta; \mathbf{w}_j\}$ ▷ atualiza o conjunto de valores conhecidos da função objetivo
12: **return** $\boldsymbol{\theta}^\star \leftarrow \underset{\boldsymbol{\theta}}{argmax}\ \mathbf{W}_\Theta$ ▷ Retorna o melhor ponto em $\boldsymbol{\Theta}$.

6.3 Algumas Observações Sobre a Sintonia do UKF

É importante observar que a maximização da verossimilhança de uma quantidade limitada de dados de treinamento $\mathbf{Y}$ não garante a maximização da qualidade das estimativas de estado produzidas pelo filtro, pois não são utilizadas informações a respeito dos estados reais que deram origem a essas séries temporais [32]. Seria muito mais conveniente maximizar a verossimilhança dos estados reais que deram origem às medidas. No entanto, há casos em que seria custoso obter estados reais em quantidade suficiente para realizar o processo de sintonia (ver, por exemplo, os experimentos relatados em [32]). Assim, é importante desenvolver estratégias de sintonia do UKF que dependam somente das medidas passíveis de serem obtidas.

Ainda que fosse possível ter acesso aos estados reais que deram origem às séries de treinamento, a maximização da qualidade das predições do UKF em um conjunto limitado de dados não garantiria que o filtro apresentaria desempenho adequado em todas as situações que viesse a encontrar. Como em qualquer processo de ajuste de modelos paramétricos, a qualidade dos resultados obtidos pelo processo de sintonia do UKF depende essencialmente do quão bem os dados disponíveis para treinamento representam as situações passíveis de serem encontradas pelo filtro.

Outro ponto a ser lembrado está relacionado com sistemas dinâmicos variantes no tempo. Para sistemas dinâmicos que mudam rapidamente com o tempo, uma abordagem *online* seria mais adequada. Se, porém, o sistema dinâmico mudar lentamente com o tempo, o custo computacional adicional imposto pelas abordagens de sintonia *online* existentes pode não ser justificável. Nesse caso, poderia ser mais interessante refazer a sintonia *offline* de tempos em tempos. Por completude, os experimentos numéricos descritos nos Capítulos 9, 10 e 11 incluem o UKF sintonizado com a estratégia de adaptação *online* descrita em [1].

Capítulo 7

Algoritmos para Sintonia do UKF

No Capítulo 6, o processo de sintonia do UKF foi formulado como um problema de otimização de uma função caixa preta. Neste capítulo, apresentamos dois algoritmos que implementam esse processo de sintonia.

O primeiro é um algoritmo de busca estocástica [40], baseado no BSF [9]. É um algoritmo de fácil implementação e de baixa complexidade computacional. Em teoria, porém, algoritmos de busca estocástica podem demandar a realização de um número significativo de avaliações da função objetivo, o que poderia dificultar sua utilização em situações nas quais o custo de avaliação da função objetivo fosse significativo.

Em razão disso, o segundo algoritmo é baseado na teoria da otimização por modelos substitutos [39], a qual é voltada especificamente para situações nas quais a avaliação da função objetivo é significativamente custosa. Em contrapartida, o segundo algoritmo apresenta maior complexidade computacional que o primeiro.

7.1 Sintonia do UKF Inspirada no Filtro de *Bootstrap*

7.1.1 Abordando a Sintonia do UKF como um Problema de Otimização Estocástica

Vimos no Capítulo 6 que a sintonia dos parâmetros do UKF pode ser vista como um problema de otimização de uma função caixa preta. Considerando que o logaritmo da verossimilhança marginal (6.2), peça central da função objetivo adotada, é função não linear dos parâmetros do UKF, a função objetivo adotada pode não permitir o uso eficiente de algoritmos baseados em gradiente, embora, em princípio, informações de derivada pudessem ser obtidas por aproximações numéricas. Ela poderá conter irregularidades, tais como pontos de sela e descontinuidades de salto, e poderá exibir múltiplos máximos locais. Essas características dificultariam a utilização de algoritmos baseados em derivadas.

Uma abordagem mais geral para a realização da sintonia consistiria em utilizar algoritmos de otimização livres de gradiente, tais como *pattern search* [41] ou algoritmos genéticos [42]. O problema é que tais algoritmos usualmente demandam a realização de um número significativo de avaliações da função objetivo, e a função definida pelo Algoritmo 1 pode ser relativamente custosa, caso, por exemplo, as séries temporais que compõem os dados de treinamento sejam longas.

Seria então interessante obter um algoritmo de baixa complexidade computacional, capaz de sintonizar satisfatoriamente os parâmetros do UKF com base em algumas dezenas de avaliações da função objetivo. Para conceber um algoritmo com essas características, assumimos

que a função objetivo descrita pelo Algoritmo 1 apresenta algum grau de continuidade, o que é uma hipótese razoável para funções de engenharia.

Para funções contínuas, os valores da função objetivo calculados em dois pontos distintos do domínio da função estão relacionados com a distância entre esses pontos. Quanto mais próximos estão os pontos, mais similares são os valores da função objetivo. Essa propriedade significa que próximo a pontos nos quais o valor da função objetivo é alto há pontos nos quais o valor da função objetivo é similarmente alto, ou ainda mais alto.

Um algoritmo de sintonia do UKF poderia então fazer uso de buscas locais para tentar acelerar a convergência e assim reduzir a quantidade de avaliações da função objetivo. O problema é que buscas locais tendem a ficar presas em máximos locais. Para minimizar essa possibilidade, o algoritmo de sintonia teria de também realizar algum tipo de exploração do domínio da função objetivo.

Na Seção 7.1.2, propomos um algoritmo de sintonia que busca balancear esses objetivos conflitantes, avaliando a função objetivo em pontos obtidos por amostragem de uma determinada distribuição de probabilidade. Essa distribuição de probabilidade seria concebida de modo a produzir tanto pontos mais próximos a pontos nos quais se sabe que o valor da função objetivo é alto (busca local) como pontos mais distantes (exploração do domínio).

7.1.2 Sintonia Baseada em Partículas

O algoritmo proposto é baseado em ideias do BSF. O BSF aproxima a distribuição de probabilidade do vetor de estado $\mathbf{x}_k$[1] no instante de tempo k por meio de um conjunto de amostras aleatórias (partículas). A média e a covariância da estimativa de estado $\hat{\mathbf{x}}_k$ produzida pelo filtro no instante k são dadas respectivamente pela média e covariância das partículas existentes no instante k. O filtro funciona sequencialmente, executando alternadamente um passo de predição e um passo de correção.

A fim de descrever brevemente o funcionamento do BSF quando aplicado ao sistema dinâmico (3.5), vamos assumir que, no instante de tempo $k-1$, a população de partículas seja $\mathbf{X}_{k-1} = \{\mathbf{x}_{k-1}(i) : i = 1, \ldots, N\}$, onde $\mathbf{x}_{k-1}(i)$ é a i-*ésima* partícula em $\mathbf{X}_{k-1}$. Esse conjunto de partículas é de fato um conjunto de amostras aleatórias da distribuição de probabilidade $p(\mathbf{x}_{k-1}|\mathbf{D}_{k-1})$, onde $\mathbf{D}_{k-1} = \{\mathbf{y}_i, i = 1, \ldots, k-1\}$ é o conjunto de medidas disponível em $k-1$.

O passo de predição objetiva predizer qual seria o próximo estado $\mathbf{x}_k \sim p(\mathbf{x}_k|\mathbf{D}_k)$. Para tanto, cada partícula em $\mathbf{X}_{k-1}$ é transformada pela função de transição de estado $f(\cdot)$, produzindo amostras prévias $\mathbf{x}_k^-(i) = f(\mathbf{x}_{k-1}(i)) + \mathbf{w}_{k-1}$, com $\mathbf{w}_{k-1}$ obtido por amostragem da distribuição de probabilidade do ruído de processo.

O passo de atualização utiliza a medida $\mathbf{y}_k$ para corrigir as amostras prévias geradas no passo de predição. Para isso, atribui pesos normalizados às amostras prévias, de modo que o peso q_i atribuído à amostra prévia $\mathbf{x}_k^-(i)$ seja proporcional à verossimilhança condicional $p(\mathbf{y}_k|\mathbf{x}_k^-(i))$. Cada peso q_i é visto como a massa de probabilidade associada com a amostra prévia $\mathbf{x}_k^-(i)$ e é usado para formar uma distribuição de probabilidade discreta sobre as amostras prévias. As amostras (partículas) corrigidas $\mathbf{x}_k(i)$ são então obtidas por amostragem dessa distribuição de probabilidade discreta.

As operações realizadas pelo algoritmo de sintonia aqui proposto se assemelham a algumas das operações realizadas no passo de correção do BSF. A cada iteração do algoritmo proposto, as partículas associadas com os N_s maiores valores da função objetivo são selecionadas, formando um subconjunto $\mathbf{\Theta}^\star$. Os valores da função objetivo correspondentes às partículas em $\mathbf{\Theta}^\star$ são então normalizados para o intervalo $[0, 1]$, de modo que os pontos em

[1] Quando falamos em estados e medidas, estamos sempre nos referindo ao sistema dinâmico dado por (3.5)

$\boldsymbol{\Theta}^\star$ e os pesos normalizados formam uma distribuição de probabilidades discreta. Uma população $\boldsymbol{\Theta}_{best}$ é então formada amostrando-se aleatoriamente N_s partículas dessa distribuição discreta. [2]

Esse processo de reamostragem tem como efeito criar populações nas quais o número de cópias das partículas com pesos mais altos tende a ser maior que o número de cópias das partículas com pesos mais baixos. Esse efeito pode ser visto como uma maneira de fazer uso do que já se sabe a respeito da forma da função objetivo, mas a estratégia não traz novas informações para o processo de otimização. Para trazer novas informações, o algoritmo proposto amostra N_s novas partículas da distribuição

$$p(\boldsymbol{\theta}_{new}) = \mathcal{N}(\mathbf{M}_{best}, \mathbf{P}_{best}) \ , \tag{7.1}$$

onde $\mathbf{M}_{best}$ e $\mathbf{P}_{best}$ são respectivamente a média e a covariância da população $\boldsymbol{\Theta}_{best}$.[3] A função objetivo é avaliada nos pontos $\boldsymbol{\theta}_{new}$, gerando os pesos correspondentes a essas partículas. As novas partículas e os pesos correspondentes são então adicionados à população atual de partículas, trazendo assim novas informações para o processo de otimização.

A distribuição de probabilidade $\mathcal{N}(\mathbf{M}_{best}, \mathbf{P}_{best})$ é interessante porque pode produzir amostras localizadas a distâncias variadas das melhores partículas existentes. Amostras que estiverem próximas às melhores partículas existentes permitem que o algoritmo de sintonia faça uso do que já se sabe a respeito da forma da função objetivo, enquanto amostras mais distantes permitem que o algoritmo explore o domínio da função objetivo. Os passos gerais que compõem a estratégia de sintonia proposta são apresentados no Algoritmo 3. O algoritmo recebe como entradas as séries de treinamento $\mathbf{Y}$, a faixa de valores (Range) permitidos para as partículas, o número máximo de avaliações da função objetivo (budget), um conjunto $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$ de N partículas iniciais e o número N_s de novas partículas geradas a cada iteração.

[2] Nos experimentos numéricos realizados nesse trabalho, essa reamostragem é realizada de acordo com a estratégia de reamostragem estratificada [43].

[3] A distribuição normal foi escolhida porque o UKF é um filtro Gaussiano.

Algorithm 3 Sintonia Baseada em Partículas

1: **procedure** PFUKF-OPT(N_s, Budget, Range, $\boldsymbol{\Theta}$,$\mathbf{Y}$)
2: $\mathbf{W}_\Theta \leftarrow$ Usando as séries de treinamento $\mathbf{Y}$, avalie a função objetivo nas partículas $\boldsymbol{\Theta}$, gerando os pesos correspondentes.
3: $\boldsymbol{\Theta}_{best} \leftarrow \boldsymbol{\Theta}$
4: **for** $i = 1$ to $\frac{\text{Budget}}{N_s}$ **do**
5: $\mathbf{M}_{best} \leftarrow$ Calcular a média das partículas em $\boldsymbol{\Theta}_{best}$.
6: $\mathbf{P}_{best} \leftarrow$ Calcular a covariância das partículas em $\boldsymbol{\Theta}_{best}$.
7: $\boldsymbol{\Theta}_{new} \leftarrow$ Amostrar N_s partículas de $\mathcal{N}(\mathbf{M}_{best}, \mathbf{P}_{best})$.
8: $\boldsymbol{\Theta}_{new} \leftarrow \text{trim}(\boldsymbol{\Theta}_{new}, Range)$ ▷ Os valores das coordenadas das partículas $\boldsymbol{\Theta}_{new}$ são limitados aos valores máximos definidos em Range.
9: $\mathbf{W}_{new} \leftarrow$ Usando as séries de treinamento $\mathbf{Y}$, avalie a função objetivo nas novas partículas $\boldsymbol{\Theta}_{new}$, gerando os pesos correspondentes.
10: $\boldsymbol{\Theta} \leftarrow \{\boldsymbol{\Theta}; \boldsymbol{\Theta}_{new}\}$ ▷ Adicione as novas partículas às partículas existentes $\boldsymbol{\Theta}$.
11: $\mathbf{W}_\Theta \leftarrow \{\mathbf{W}_\Theta; \mathbf{W}_{new}\}$ ▷ Adicione os novos pesos aos pesos existentes $\mathbf{W}_\Theta$.
12: $\mathbf{W}_\Theta \leftarrow$ Manter somente os N_s melhores pesos em $\mathbf{W}_\Theta$.
13: $\boldsymbol{\Theta} \leftarrow$ Manter somente as partículas correspondentes aos pesos em $\mathbf{W}_\Theta$.
14: $\boldsymbol{\Theta}_{best} \leftarrow$ Reamostrar N_s partículas da distribuição de probabilidades discreta formada com $\boldsymbol{\Theta}$ e $\mathbf{W}_\theta$
15: **return** $\boldsymbol{\theta}^\star \leftarrow \underset{\boldsymbol{\theta}}{argmax}\ \mathbf{W}_\Theta$ ▷ Retorna a partícula associada ao maior peso

7.2 Sintonia do UKF com o Auxílio de Modelos Substitutos

7.2.1 Noções de Otimização por Modelos Substitutos

A otimização baseada em modelos [39] é geralmente utilizada em situações nas quais o custo de avaliação da função objetivo $g(\cdot)$ é significativo. A estratégia geral consiste em utilizar amostras da função objetivo obtidas em um dado número de pontos do domínio da função, para construir um modelo computacional que é então usado no lugar da função real no processo de otimização.

O modelo da função objetivo é usualmente baseado em meta modelos probabilísticos, tais como Processos Gaussianos (GP) [44]. O modelo da função é obtido usando-se o GP como modelo de regressão para um conjunto de dados $\mathcal{D} = \{(\boldsymbol{\theta}_1, g(\boldsymbol{\theta}_1)), \ldots, (\boldsymbol{\theta}_{n_D}, g(\boldsymbol{\theta}_{n_D}))\}$ com n_D amostras, onde $g(\boldsymbol{\theta}_i)$ é o valor da função objetivo no ponto $\boldsymbol{\theta}_i$. Uma vez treinado, o modelo GP pode ser usado para predizer o valor da função objetivo em pontos do domínio que não pertencem a $\mathcal{D}$. Para fazer predições, GPs assumem que a função objetivo é contínua. Desse modo, se dois pontos do domínio estiverem próximos um do outro, os valores da função objetivo calculados nesses pontos também deverão estar próximos.

Para equacionar a relação entre a distância entre dois pontos e a similitude dos valores da função objetivo nesses pontos, o paradigma GP utiliza funções de covariância. Uma função de covariância simples, mas que ilustra bem esse conceito, é a *Exponencial Quadrática* (SE), a qual é dada por

$$\text{cov}(g(\boldsymbol{\theta}_i), g(\boldsymbol{\theta}_j)) = \exp\left(-\frac{1}{2}|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j|^2\right) . \tag{7.2}$$

Para a função SE, a covariância entre os valores da função objetivo em dois pontos do

domínio se aproxima de 1, na medida que a distância entre esses pontos tende a zero, e se aproxima de zero, na medida em que a distância cresce. As várias outras funções de covariância utilizadas no contexto dos GPs funcionam segundo o mesmo princípio.

Modelos GP utilizam essa propriedade para associar covariâncias às próprias predições. Quanto mais próximo o ponto no qual se realiza a predição estiver de pontos do conjunto $\mathcal{D}$ (pontos utilizados no treinamento do modelo), mais confiante é a predição, e menor é a covariância associada ao valor predito pelo modelo. A covariância associada à predição é de fundamental importância, pois é usada no processo de refinamento sequencial do modelo, que descrevemos a seguir.

Dentro do paradigma da Otimização Sequencial Baseada em Modelos (SMBO), o refinamento é realizado de forma concomitante à otimização. Em linhas gerais, o algoritmo de otimização usa as predições do modelo para encontrar o ponto $\boldsymbol{\theta}_l$ do domínio da função objetivo que maximiza um dado critério heurístico, usualmente denominado critério/função de aquisição. A função objetivo é avaliada em $\boldsymbol{\theta}_l$, e o par resultante $(\boldsymbol{\theta}_l, g(\boldsymbol{\theta}_l))$ é adicionado aos dados de treinamento $\mathcal{D}$. O modelo é ajustado ao novo conjunto $\mathcal{D}$, e o otimizador usa o modelo atualizado para novamente buscar por um ponto onde avaliar a função objetivo, reiniciando o processo. O processo de otimização/refinamento pode ser finalizado de maneiras diversas, sendo comum a definição de um número máximo de avaliações da função objetivo. Os passos gerais dos algoritmos SMBO, segundo [45], são mostrados no Algoritmo 4.

Algorithm 4 Passos Gerais dos Algoritmos SMBO

1: **procedure** SMBO
2: Use os dados de treinamento $\mathcal{D}$ para produzir um modelo estocástico da função objetivo (o modelo substituto).
3: Use o modelo substituto para buscar por um ponto no domínio da função objetivo que maximiza um determinado critério de aquisição.
4: Avalie a função objetivo no ponto encontrado no passo 3.
5: Atualize os dados de treinamento $\mathcal{D}$ usando o novo ponto e o valor correspondente da função objetivo.
6: Atualize o modelo substituto utilizando os dados de treinamento $\mathcal{D}$.
7: Se o critério de parada ainda não foi atendido, retorne ao passo 3.

Critérios de Aquisição

Há uma quantidade significativa de critérios de aquisição [46] que podem ser usados na implementação do Passo 3 do Algoritmo 4. Esses critérios usualmente se valem do fato de que as predições geradas pelo modelo substituto são compostas por um valor médio (o valor predito) e por uma covariância. Esta é utilizada para balancear as estratégias conflitantes de buscar próximo a pontos que exibem valores altos da função objetivo (busca local) e buscar em regiões do domínio da função objetivo nas quais foram realizadas poucas (ou nenhuma) avaliações da função objetivo (busca exploratória). Como vimos na Seção 7.1, a busca local pode acelerar a convergência, mas tende a ficar presa em máximos locais, por isso é necessário avaliar a função objetivo em regiões pouco exploradas do seu domínio.

Um critério de aquisição que explicitamente tenta equilibrar buscas locais e buscas exploratórias é o *Upper Confidence Bound* (UCB) ([47], [48]). O valor do UCB em um ponto $\boldsymbol{\theta}$ é dado por

$$\mathrm{UCB}(\boldsymbol{\theta}) = \mu(\boldsymbol{\theta}) + A\sigma(\boldsymbol{\theta}) \ , \tag{7.3}$$

onde $A > 0$ é um valor escalar, enquanto $\mu(\boldsymbol{\theta})$ e $\sigma(\boldsymbol{\theta})$ são respectivamente a média e o desvio padrão da predição gerada pelo modelo substituo no ponto $\boldsymbol{\theta}$. É sabido que o valor do parâmetro livre A pode influenciar significativamente os resultados da otimização [11]. Embora [49] tenha proposto um método para determinar automaticamente esse valor, ainda é comum escolher $A = 2$.

Outro critério de aquisição que equilibra buscas locais e buscas exploratórias é o *Expected Improvement* (EI) [45]. Esse critério estima o quanto de melhora sobre o melhor valor conhecido da função objetivo poderia ser obtida, se a função objetivo fosse avaliada em um dado ponto $\boldsymbol{\theta}$ do domínio da função. Para maximização, o EI é dado por

$$\mathrm{EI}(\boldsymbol{\theta}) = \begin{cases} (\mu(\boldsymbol{\theta}) - g_{max})\Phi\left(\frac{\mu(\boldsymbol{\theta})-g_{max}}{\sigma(\boldsymbol{\theta})}\right) + \sigma(\boldsymbol{\theta})\phi\left(\frac{\mu(\boldsymbol{\theta})-g_{max}}{\sigma(\boldsymbol{\theta})}\right) & \text{se } \sigma(\boldsymbol{\theta}) > 0 \\ 0 & \text{se } \sigma(\boldsymbol{\theta}) = 0 \end{cases} , \tag{7.4}$$

onde $\mathrm{EI}(\boldsymbol{\theta})$ é a melhora esperada em $\boldsymbol{\theta}$, g_{max} é o valor mais alto da função objetivo encontrado em $\mathcal{D}$, Φ é a função distribuição cumulativa normal e ϕ é a função densidade de probabilidade normal. Vale ressaltar que o EI não apresenta parâmetros a serem definidos pelo usuário.

7.2.2 Algoritmo de Sintonia Baseado em Modelos

Os algoritmos SMBO fazem uso de diversos algoritmos de otimização para encontrar o ponto do domínio que, segundo o modelo substituto, maximiza o critério de aquisição escolhido (Passo 3 do Algoritmo 4). A maximização do critério de aquisição deveria ser global, no sentido de que a busca deveria cobrir todo o domínio da função objetivo. Para domínios de duas ou mais dimensões, essa tarefa só pode ser realizada de maneira aproximada.

A maximização do critério de aquisição é então usualmente implementada com otimizadores globais heurísticos, como algoritmos genéticos [42] ou *simulated annealing* [50]. Por serem heurísticos, há sempre a chance de que esses métodos não encontrem o ponto que maximiza o critério de aquisição. Em razão disso, o otimizador aqui descrito propõe implementar buscas locais de modo simples, com o intuito de complementar as buscas globais realizadas. Em poucas palavras, a estratégia do sintonizador proposto consiste em efetuar a busca global e a busca local, calcular o critério de aquisição escolhido nos pontos obtidos pelas buscas, e avaliar a função objetivo no ponto que produziu o maior valor do critério de aquisição.

Para implementar a busca global, o sintonizador proposto faz uso do comumente utilizado algoritmo genético. Desse modo, o usuário pode utilizar qualquer critério de aquisição que não utilize derivadas.

A busca local é implementada calculando o critério de aquisição em N_s pontos locais $\boldsymbol{\theta}_{local}$ gerados de acordo com a distribuição de probabilidade

$$p(\boldsymbol{\theta}_{local}) = \mathcal{N}(\mathbf{M}_{best}, \mathbf{S}_{best}) , \tag{7.5}$$

onde $\mathbf{M}_{best}$ e $\mathbf{S}_{best}$ são respectivamente a média e a covariância de $\boldsymbol{\Theta}_{best}$, uma população formada por cópias dos melhores pontos de treinamento, e N_s é um inteiro positivo definido pelo usuário. Vejamos agora como a população $\boldsymbol{\Theta}_{best}$ é formada e por que efetuar busca local em amostras de (7.5) poderia ter efeito positivo na maximização do critério de aquisição (passo 3 do Algoritmo 4).

Em cada iteração do otimizador proposto, os pontos do conjunto de treinamento $\mathcal{D}$ associados com os N_s valores mais altos da função objetivo são selecionados, formando um subconjunto $D^\star$. Os valores da função objetivo associadas com os pontos em $D^\star$ são então

normalizados para o intervalo $[0, 1]$, de modo que os pontos em $D^\star$ e os valores normalizados correspondentes formem uma distribuição de probabilidades discreta. A população $\boldsymbol{\Theta}_{best}$ é então formada amostrando-se aleatoriamente N_s pontos dessa distribuição. Nos experimentos numéricos descritos nesse documento, essa amostragem é implementada de acordo com o procedimento de reamostragem estratificada [43], comumente adotado em filtros de partículas [26].

Pontos associados a valores normalizados mais altos têm assim mais chances de serem reamostrados do que pontos associados a valores mais baixos. Essa característica pode ser vista como uma maneira de utilizar a informação contida nos pontos de treinamento, mas a estratégia não traz novas informações para o processo de otimização. Sem novas informações, o processo de reamostragem faria com que a população $\boldsymbol{\Theta}_{best}$ terminasse sendo formada somente por cópias do ponto de treinamento associado ao maior valor normalizado.

É para trazer novas informações para o processo de otimização que o algoritmo de sintonia proposto amostra N_s novos pontos da distribuição (7.5) (a distribuição normal foi escolhida porque o UKF é um filtro Gaussiano). Essa estratégia é interessante, porque pode gerar amostras localizadas a distâncias variadas dos pontos na população $\boldsymbol{\Theta}_{best}$. Amostras que estiverem mais próximas a esses pontos permitiriam ao algoritmo fazer uso da informação existente a respeito da forma da função objetivo, enquanto pontos mais distantes permitiriam explorar o espaço de parâmetros. A questão agora é se haveria chance de o critério de aquisição ter valor mais alto em um ponto local $\boldsymbol{\theta}_{local}$ do que em um ponto encontrado por busca global implementada por estratégia heurística, como, por exemplo, um algoritmo genético.

Como vimos, o valor da predição produzida por um modelo GP em um dado ponto está correlacionado com os valores da função objetivo nos pontos de treinamento localizados próximos ao ponto de predição. Como resultado, predições feitas nos pontos locais $\boldsymbol{\theta}_{local}$ provavelmente apresentariam valores altos, o que aumentaria as chances de o critério de aquisição ser mais alto do que em um ponto encontrado por busca global heurística. Quando isso acontece, a busca local foi capaz de contribuir para o passo 3 do Algoritmo 4.

O sintonizador proposto é detalhado no Algoritmo 5. Suas entradas são as séries de treinamento $\mathbf{Y}$, a faixa de valores (Range) permitidos para as partículas, o número máximo de avaliações da função objetivo (budget), um conjunto $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$ de N pontos iniciais e o número N_s de candidatos locais gerados a cada iteração.

Algorithm 5 Sintonia Baseada em Modelos

1: **procedure** UKF OPT($\mathbf{Y}$,N_s, $\mathbf{\Theta}$, **Range**, **budget**)
2: $\mathbf{W}_\Theta \leftarrow$ Usando os dados de treinamento $\mathbf{Y}$, calcula a função objetivo nos pontos de treinamento em $\mathbf{\Theta}$
3: $\mathcal{M} \leftarrow$ Treina o modelo GP em $\mathcal{D} = \{\mathbf{\Theta}, \boldsymbol{W}_\theta\}$
4: **while** $nevals < budget$ **do** ▷ $nevals$ - número de avaliações realizadas da função objetivo
5: $\mathbf{\Theta}_{best} \leftarrow$ Reamostragem dos N_s pontos de $\mathbf{\Theta}$ associados aos maiores valores em $\mathbf{W}_\Theta$
6: $\mathbf{M}_{best} \leftarrow$ Calcula a média de $\mathbf{\Theta}_{best}$
7: $\mathbf{P}_{best} \leftarrow$ Calcula a matriz de covariância de $\mathbf{\Theta}_{best}$
8: $\mathbf{\Theta}_{local} \leftarrow \left\{\boldsymbol{\theta}_{local}^{(1)}, \boldsymbol{\theta}_{local}^{(2)}, \ldots, \boldsymbol{\theta}_{local}^{(N_s)}\right\}$ ▷ Amostra N_s pontos de $\mathcal{N}(\mathbf{M_{best}}, \mathbf{P_{best}})$
9: $\boldsymbol{\phi}_{local}^\star \leftarrow$ Seleciona o ponto local que produz o maior valor do critério de aquisição
10: $\boldsymbol{\phi}_{global}^\star \leftarrow$ Efetua busca global pelo ponto que maximiza o critério de aquisição
11: **if** critério de aquisição em $\boldsymbol{\phi}_{local}^\star$ maior que em $\boldsymbol{\phi}_{global}^\star$ **then**
12: $\boldsymbol{\theta}_{new} \leftarrow \boldsymbol{\phi}_{local}^\star$ ▷ Seleciona o ponto local
13: **else**
14: $\boldsymbol{\theta}_{new} \leftarrow \boldsymbol{\phi}_{global}^\star$ ▷ Seleciona o ponto global
15: $w_{new} \leftarrow$ Usando os dados de treinamento $\mathbf{Y}$, avalia a função objetivo em $\boldsymbol{\theta}_{new}$.
16: $\mathbf{\Theta} \leftarrow \{\mathbf{\Theta}; \boldsymbol{\theta}_{new}\}$ ▷ Adiciona $\boldsymbol{\theta}_{new}$ a $\mathbf{\Theta}$
17: $\mathbf{W}_\Theta \leftarrow \{\mathbf{W}_\Theta; w_{new}\}$ ▷ Adiciona w_{new} a $\mathbf{W}_\Theta$
18: $\mathcal{M} \leftarrow$ Treina o modelo GP em $\mathcal{D} = \{\mathbf{\Theta}, \boldsymbol{W}_\theta\}$
19: **return** $\boldsymbol{\theta}^\star \leftarrow \underset{\boldsymbol{\theta}}{argmax}\ \mathbf{W}_\Theta$ ▷ Retorna o ponto associado ao maior peso

Os passos 9 e 10 do Algoritmo 5 devem assegurar que os pontos gerados permaneçam nos limites definidos por **Range**. Há mais de uma maneira de isso ser feito. Para o critério UCB (7.3), uma heurística que funcionou bem nos experimentos numéricos consistiu em penalizar o valor do critério de aquisição em função dos valores das coordenadas do ponto. Para um dado ponto $\boldsymbol{\theta}$, a penalização é dada por $\sum_{i=1}^{n_\theta}(\theta_i^{(c)})^{20}$, onde $\boldsymbol{\theta}^{(c)}$ é o resultado da conversão de $\boldsymbol{\theta}$ do intervalo definido por **Range** para o intervalo $[-1, 1]^{n_\theta}$. Uma estratégia similar é adotada em [36]. Para o critério EI (7.4), os melhores resultados foram obtidos limitando-se os valores das coordenadas do ponto, de modo que os valores máximo e mínimo dessas coordenadas sejam os valores dados por **Range**.

Complexidade Computacional

A complexidade computacional dos modelos GP é $\mathcal{O}(n^3)$ para treinamento e $\mathcal{O}(n^2)$ para predição, onde n é o número de pontos de treinamento [51]. Se n for grande, na casa de algumas centenas, essas complexidades implicariam que os custos computacionais associados ao Algoritmo 5 seriam significativos. Se fosse esse o caso, tais custos poderiam ser diminuídos com a adoção de alternativas como o uso de GPs esparsos [52]. Para a sintonia do UKF, porém, o número de pontos de treinamento utilizados nos experimentos numéricos foi baixo, permitindo executar o Algoritmo 5 em questão de poucos minutos em um notebook absolutamente comum.

7.2.3 Abordagens Relacionadas

Outros algoritmos de otimização baseados em modelos substitutos fazem uso de buscas locais e globais. Um algoritmo bastante sofisticado que combina buscas locais e globais é descrito em [53]. A ideia básica desse algoritmo é usar modelos GP e o critério de aquisição EI para escolher pontos no domínio da função objetivo que servirão de pontos de início para otimizadores locais do tipo *pattern search* [41] que implementam buscas concorrentes pelo ponto de máximo da função objetivo. Esse algoritmo, porém, é de implementação bastante complexa e computacionalmente caro.

Outro ponto a ser citado é que o sintonizador descrito no Algoritmo 5 é significativamente diferente daquele descrito em ([38],[36]), que é o Filtro de Kalman *unscented* sintonizado de acordo com [5] (UkfO). O UkfO utiliza um otimizador baseado em derivadas para implementar o Passo 3 do Algoritmo 4. Com isso, está limitado a usar o critério de aquisição UCB, pois este pode ser derivado. O otimizador aqui proposto implementa esse passo por meio da combinação de buscas globais e locais. Como a busca global é implementada por um algoritmo genético, qualquer critério de aquisição que não dependa de informação de gradiente pode ser usado.

7.3 Sintonia de Todos os Parâmetros do UKF

Conforme já vimos, os algoritmos 3 e 5 podem ser usados para sintonizar todos os parâmetros do UKF, incluindo as matrizes de covariância dos ruídos de processo e de medida. Para evitar aumentar demais o número de dimensões do domínio da função objetivo, consideramos aqui que as matrizes de covariância dos ruídos de processo e de medida são diagonais.

Vale lembrar que uma solução recente para o problema de definição dessas matrizes é dada pelo filtro GP-UKF([54], [55]), mas a solução demanda a existência de dados de treinamento que incluem o estado real do sistema dinâmico, além de o GP-UKF ser significativamente lento para efetuar predições.

Capítulo 8

Sobre as Simulações Numéricas

Este capítulo apresenta a moldura geral dos experimentos numéricos[1] realizados nos Capítulos 9, 10 e 11. São apresentados os filtros utilizados, os procedimentos adotados na geração de dados do sistema dinâmico, os critérios de medida de desempenho e os problemas de filtragem.

8.1 Os Filtros Utilizados

Essa seção descreve brevemente os filtros usados nos experimentos numéricos. Aos filtros abaixo descritos somam-se o Filtro de Kalman *unscented* sintonizado pelo Algoritmo 3 (PfUkf) e o Filtro de Kalman *Unscented* sintonizado pelo Algoritmo 5 (UkfA).

8.1.1 O Filtro de Kalman Estendido

O EKF [13] foi incluído na comparação porque tem sido o filtro não linear mais popular das últimas quatro décadas [56]. Parte dessa popularidade certamente se deve ao baixo custo computacional do filtro.

É sabido, porém, que o fato de o filtro aproximar as não linearidades do modelo do sistema dinâmico por meio de expansões em série de Taylor de ordem baixa pode dificultar sua aplicabilidade em situações caracterizadas por não linearidades significativas. Mais especificamente, a qualidade das estimativas de estado produzidas pelo EKF pode ser muito baixa, caso o sistema dinâmico apresente não linearidades pronunciadas. Uma exposição detalhada das limitações do EKF pode ser encontrada em [8].

Nos experimentos numéricos realizados neste trabalho, utilizamos o EKF de primeira ordem, que utiliza expansões em série de Taylor de primeira ordem. A descrição detalhada do EKF de primeira ordem pode ser encontrada na Seção 10.3 de [21].

8.1.2 UKF Padrão

O UKF tem sido adotado como alternativa superior ao EKF em problemas caracterizados por não linearidades significativas. A descrição apresentada na Seção 4.2 mostra que o UKF demanda a sintonia dos 3 parâmetros adimensionais da UT e, possivelmente, das matrizes de covariância dos ruídos de processo e de medida.

O Filtro de Kalman *Unscented* sintonizado de acordo com [4] (UkfD) é obtido quando as covariâncias dos ruídos são conhecidas e a sintonia dos parâmetros da UT é dada por $\alpha = 1$, $\beta = 0$ e $\kappa = 3 - n_x$, onde n_x é o número de dimensões do vetor de estado.

[1] As simulações numéricas foram implementadas com o Matlab Student Edition R2013a.

8.1.3 Filtro de Kalman de Cubatura

O CKF é um filtro Gaussiano que foi descrito por seus autores como sendo numericamente mais preciso e mais estável que o UKF (veja a Seção VII de [57]). É sabido, porém, que o CKF é um caso particular do UKF, obtido quando a sintonia da UT é dada por $\alpha = 1$, $\beta = 0$ and $\kappa = 0$ [36]. O principal fator que levou à inclusão do CKF nessa comparação foi exatamente o fato de esse (relativamente) popular filtro poder ser obtido por meio de uma sintonia específica do UKF.

8.1.4 UKF com Adaptação Online do Parâmetro de Escalamento da UT

O Filtro de Kalman *Unscented* sintonizado de acordo com ([1], [2], [3]) (Ukf-ML) implementa estratégia de sintonia *online* da UT. Assumindo que as covariâncias dos ruídos são conhecidas, o valor do parâmetro κ da UT é escolhido de um conjunto discreto de valores possíveis, cada vez que o filtro recebe uma medida, de acordo com um determinado critério de desempenho.

Vários critérios de desempenho são propostos pelos autores, incluindo a maximização da verossimilhança condicional da medida recebida, $p(\mathbf{y}_k|\kappa)$. Para um dado valor de κ, essa verossimilhança é aproximada por

$$p(\mathbf{y}_k|\kappa) \approx \mathcal{N}\{\mathbf{y}_k|\hat{\mathbf{y}}_k(\kappa), \mathbf{P}_k(\kappa)\} \tag{8.1}$$

onde $\hat{\mathbf{y}}_k(\kappa)$ e $\mathbf{P}_k(\kappa)$ são simplesmente os dois primeiros momentos da medida predita pelo UKF sintonizado com o valor de κ em questão. Os parâmetros restantes da UT são mantidos em $\alpha = 1$ and $\beta = 0$. Nos experimentos numéricos, κ é escolhido do conjunto $K = \{0, 0.1, 0.2, \ldots, 4.9, 5\}$.

8.1.5 UKF Obtido Por Otimização Baseada em Modelos

O último filtro utilizado é o UkfO, que, como vimos, implementa otimização *offline* do UKF, utilizando para isso o algoritmo de sintonia proposto em [36]. Foi utilizada a implementação Matlab que os próprios autores do UkfO disponibilizam em [5].

8.2 Geração dos Dados do Sistema Dinâmico

Os dados necessários para realização dos experimentos foram gerados por meio de simulações de Monte Carlo das equações do sistema dinâmico em questão. Em todos os experimentos numéricos, os dados de treinamento utilizados na sintonia dos filtros UkfA, PfUkf e UkfO são diferentes dos dados usados nos testes dos filtros.

8.3 Critérios de Desempenho

A qualidade das estimativas produzidas pelos diferentes filtros é avaliada por meio dos critérios Erro Médio Quadrático (MSE), Índice de não-confiabilidade (NCI) [58] e Negativo do Logaritmo da Verossimilhança (NLL).

Antes de ver como são calculados esses critérios, vamos definir que, se $\mathbf{z}_k \in \Re^{n_z}$ é o valor real da grandeza sendo estimada no instante k e $\hat{\mathbf{z}}_k$ é a estimativa correspondente produzida por um dado filtro, o erro de estimação é dado por

$$\mathbf{e}_k = \hat{\mathbf{z}}_k - \mathbf{z}_k \ . \tag{8.2}$$

8.3.1 MSE

Considerando (8.2), o MSE é dado por

$$\text{MSE} = \text{E}[\mathbf{e}_k \mathbf{e}_k^T] \ . \tag{8.3}$$

Se $n_z = 1$, (8.3) será um escalar, caso contrário, será uma matriz, a *matriz MSE* do erro de estimação.

Nos experimentos, foram calculados tanto o Erro Médio Quadrático nas Estimativas de Estado (MSEx) como o Erro Médio Quadrático nas Estimativas de Medida (MSEy). A princípio, quanto menor for o valor do MSE, melhor seriam as estimativas do filtro.

Vale notar, porém, que as estimativas produzidas por dois filtros podem apresentar valores MSE parecidos, mas serem distribuídas de forma significativamente diferentes. Para compensar qualquer distorção de interpretação que possa ser induzida por essa característica, os valores MSEx apresentados nesse trabalho são sempre associados a intervalos de confiança e a gráficos do tipo *box plot*, que permitem inspecionar visualmente as distribuições dos erros quadráticos dos diferentes filtros.

8.3.2 Box Plots

Em um gráfico *box plot*, é possível inspecionar visualmente a tendência central, a dispersão e a assimetria da distribuição de probabilidades de um conjunto de dados.

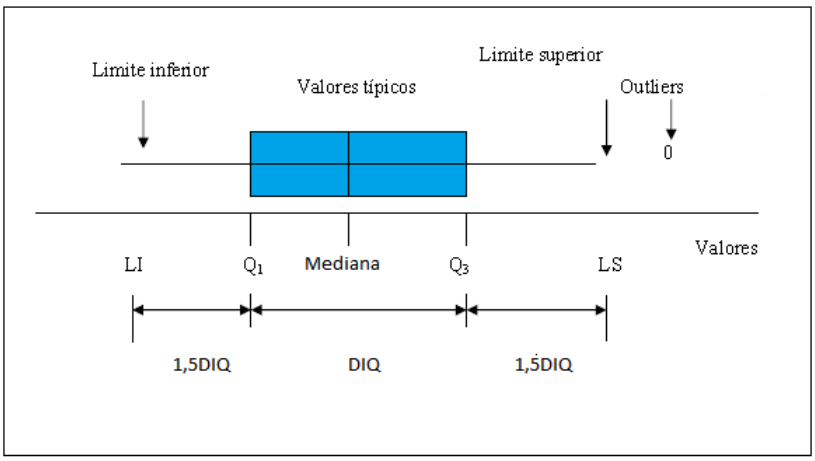

Figura 8.1: *Detalhando o Box Plot*

A tendência central da distribuição é representada pela mediana, que é o valor que divide o conjunto de dados em duas partes iguais, ou seja, 50% dos dados ficam abaixo da mediana e 50% ficam acima. Uma propriedade interessante da mediana é que ela é menos sensível aos valores extremos do conjunto de dados que a média aritmética. Assim, a mediana complementa a informação representada pelo MSE.

Os quartis dividem um conjunto de dados em quatro partes iguais. Abaixo do primeiro quartil, Q_1, estão os 25% menores valores do conjunto de dados, enquanto abaixo do terceiro quartil, Q_3, encontram-se os 75% menores valores. O desvio inter-quartílico (DIQ), que engloba 50% dos valores da distribuição, é dado pela diferença entre os valores de Q_3 e de

Q_1. Quanto maior for essa diferença, mais dispersa é a distribuição e, consequentemente, maior é a variabilidade.

Os pontos localizados entre os limites inferior (LI) e superior (LS) são considerados típicos da distribuição. Pontos localizados abaixo de $Q_1 - 1.5DIQ$ ou acima de $Q_3 + 1.5DIQ$ são considerados atípicos (*outliers*).

A simetria da distribuição é representada pela posição da mediana em relação ao primeiro e terceiro quartis. Quanto mais a mediana estiver próxima de um desses quartis, mais assimétrica é a distribuição. A simetria é importante, porque fornece indício de o quanto a distribuição de dados se distancia da curva Gaussiana.

8.3.3 NCI

Uma característica bastante útil dos filtros de Kalman em geral é a capacidade de associar graus de incerteza às estimativas que produzem. O grau de incerteza associado a uma dada estimativa indicaria ao usuário o quanto essa estimativa é confiável. A questão é que, nem sempre as incertezas produzidas pelos filtros são compatíveis com a qualidade das estimativas que produzem. Para lidar com essa questão, utilizamos o critério NCI.

O critério NCI permite avaliar o quão críveis são essas incertezas. Quanto mais críveis forem as incertezas que produz, mais confiável é considerado o filtro. É importante ressaltar que um filtro funciona com base nos dados que recebe, o que faz com que as incertezas associadas às estimativas que produz sejam dependentes desses dados. A avaliação da credibilidade das incertezas só pode então ser feita no sentido estatístico. Na prática, essa avaliação é realizada por meio de simulações de Monte Carlo.

Se o estimador estiver funcionando adequadamente, o erro de estimação (8.2) será um vetor aleatório de média zero e matriz MSE $\mathbf{\Sigma}_k$ igual à matriz de covariância $\mathbf{P}_k$ gerada pelo estimador[2]. Nesse caso ideal, teríamos um filtro/estimador perfeitamente confiável.

Na prática, assumindo que o estimador é não tendencioso, quanto mais próxima a matriz $\mathbf{P}_k$ estiver de $\mathbf{\Sigma}_k$, mais confiável será o estimador. Desse modo, a credibilidade do estimador é dada pela similitude entre as matrizes $\mathbf{P}_k$ e $\mathbf{\Sigma}_k$. O NCI provê uma maneira de quantificar essa similitude. Sendo $\mathbf{e}_k$ dado por (8.2), o NCI da k-ésima estimativa é

$$\begin{aligned} \text{NCI} &= \frac{10}{n_T}\sum_{k=1}^{n_T}\log_{10}(\rho_k) \\ \rho_k &= \frac{\mathbf{e}_k^T\mathbf{P}_k^{-1}\mathbf{e}_k}{\mathbf{e}_k^T\mathbf{\Sigma}_k^{-1}\mathbf{e}_k}\ , \end{aligned} \tag{8.4}$$

onde n_T é o número de realizações de Monte Carlo da k-ésima estimativa.

Se a hipótese de que o erro de estimação é distribuído segundo uma Gaussiana de média zero e covariância $\mathbf{\Sigma}_k$ for válida, o NCI do estimador ideal seria aproximadamente igual a zero. Valores positivos do NCI indicam que o estimador é otimista, ou seja, acredita que os erros das estimativas que produz são menores do que realmente são. Valores negativos do NCI indicam que o filtro é pessimista.

Para comparar a confiabilidade de dois ou mais filtros, basta comparar diretamente os valores absolutos dos respectivos NCI. Quanto maior esse valor, menos confiável é o filtro.

[2]Em geral, seria mais correto falar em matriz MSE gerada pelo filtro em vez de matriz de covariância. Ver subseção *Modeling Assumptions*, na Seção 10.3.1 de [21]

8.3.4 NLLy

Como vimos na Seção 6.3, a otimização do MSE das predições de medida produzidas pelo filtro não garante a otimização das estimativas de estado. Em uma situação real, porém, pode ser necessário avaliar uma determinada sintonia com base somente em dados de medidas. Assim, analisaremos a qualidade das predições de medida geradas pelos filtros com base também no Negativo do Logaritmo da Verossimilhança das predições de medida (NLLy), pois este, ao contrário do MSEy, também leva em conta as incertezas geradas pelo filtro. Para a série temporal de medidas $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$, o NLLy de uma dada sintonia $\boldsymbol{\theta}$ é

$$\text{NLLy}(\boldsymbol{\theta}) = -\frac{1}{T}\sum_{k=1}^{T} log\ \mathcal{N}(\mathbf{y}_k|\hat{\mathbf{y}}_k(\boldsymbol{\theta}), \mathbf{S}_k(\boldsymbol{\theta}))\ , \tag{8.5}$$

onde $\hat{\mathbf{y}}_k(\boldsymbol{\theta})$ e $\mathbf{S}_k(\boldsymbol{\theta})$ são respectivamente a média e a covariância da medida predita pelo filtro sintonizado com $\boldsymbol{\theta}$ no instante k. Quanto menor o NLLy, melhor a predição de medida.

8.4 Problemas de Teste

Para comparar os algoritmos de filtragem propostos no Capítulo 7, utilizamos três conhecidos problemas de filtragem não linear.

8.4.1 Kitagawa

O modelo Kitagawa[59] é um problema de filtragem desafiador. O modelo tende a estabilizar em torno de $x \pm 7$, onde é linear. O desafio da filtragem está em regiões mais distantes desse ponto, porque nelas as não linearidades se tornam significativas. As equações do modelo são

$$\begin{aligned} x_{k+1} &= 0.5x_k + \frac{25x_k}{(1+x_k^2)} + w_k\ , w_k \sim \mathcal{N}(0, 0.2^2) \\ y_k &= 5\sin(2x_k) + v_k\ , v_k \sim \mathcal{N}(0, 0.01^2)\ . \end{aligned} \tag{8.6}$$

8.4.2 Sinusoid

As equações desse sistema dinâmico são

$$x_{k+1} = 3sin(x_k) + w_k, \quad w_k \sim \mathcal{N}(0, 0.1^2) \tag{8.7}$$

$$y_k = \sigma(x_k/3) + v_k, \quad v_k \sim \mathcal{N}(0, 0.1^2)\ , \tag{8.8}$$

onde $\sigma(\cdot)$ representa o sigmóide logístico. O estado inicial obedece à distribuição $x_0 \sim \mathcal{N}(0, 1^2)$.

8.4.3 Bearings Only

Diferentes versões do problema Bearings Only são usadas na literatura ([1],[21]) para comparar o desempenho de filtros não lineares. O modelo retrata o problema de rastreamento de um alvo em movimento com base somente nas medidas de um ou mais sensores

angulares. Neste trabalho, o estado consiste na posição $\mathbf{x}_k = \{x_{1,k}, x_{2,k}\}$ no plano cartesiano. As equações desse sistema dinâmico são

$$\begin{aligned} \mathbf{x}_{k+1} &= \begin{bmatrix} 0.9 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \\ y_k &= \tan^{-1}\left(\frac{x_{2,k} - sen(k)}{x_{1,k} - cos(k)}\right) + v_k, \quad v_k \sim \mathcal{N}(0, R) \ , \end{aligned} \tag{8.9}$$

onde

$$\begin{aligned} \mathbf{Q} &= \begin{bmatrix} 0.1 & 0.01 \\ 0.01 & 0.1 \end{bmatrix} \\ R &= 0.025 \end{aligned} \tag{8.10}$$

8.5 Roteiro Geral

Os experimentos numéricos foram realizados de acordo com o Algoritmo 6.

Algorithm 6 Roteiro dos Experimentos

1: **procedure** EXPERIMENTO
2: Gerar séries de treinamento por simulação de Monte Carlo. As séries de treinamento, que são formandas somente por sequências de medidas, serão usadas para gerar as sintonias dos filtros PfUkf, UkfA e UkfO.
3: Executar o Algoritmo 3, para obter a sintonia para o filtro PfUkf.
4: Executar o Algoritmo 5, para obter a sintonia para o filtro UkfA.
5: Executar o pacote Matlab fornecido por [5], para obter a sintonia para o filtro UkfO.
6: Gerar séries de teste por simulação de Monte Carlo. As séries de teste incluem as medidas e os estados reais correspondentes.
7: Usar as séries de teste para testar todos os filtros não lineares.

Capítulo 9

Sintonia com Ruídos Conhecidos

Neste capítulo, apresentamos os resultados numéricos obtidos nos problemas de filtragem descritos no Capítulo 8. Os experimentos numéricos assumem que as matrizes de covariância reais $\mathbf{Q}$ e $\mathbf{R}$ são conhecidas. Para os Algoritmos 3 e 5, os parâmetros livres foram:

- $\mathbf{\Theta} \leftarrow 27$ amostras geradas como grade de pontos no cubo definido por Range. Para tornar a comparação mais fácil, as amostras utilizadas são aquelas geradas pelo algoritmo de sintonia do UkfO, de acordo com as funções Matlab dadas em [5]. Os valores de Range, definidos abaixo, são os valores usados em [36].
- $N_s \leftarrow 10$
- Range

  $0.01 \leq \alpha \leq 4$

  $0 \leq \beta \leq 4$

  $0 \leq \kappa \leq 5$
- budget $\leftarrow 80$

9.1 Kitagawa

9.1.1 Dados de Treinamento

O processo de otimização fez uso de uma série temporal de 1000 medidas sequenciais. A distribuição do estado inicial usada para gerar a série de treinamento foi $x_0 \sim \mathcal{N}(0, 0.5^2)$.

9.1.2 Dados de Teste

Os dados de teste consistiram de 100 séries temporais de 10 medidas cada. O número de medidas foi feito igual ao número de medidas utilizado nos experimentos descritos em [36]. A distribuição do estado inicial usada para gerar as séries de teste também foi $x_0 \sim \mathcal{N}(0, 0.5^2)$.

9.1.3 Sintonia

O processo de sintonia do UkfA usou o critério EI (7.4). A evolução dos processos de sintonia dos filtros PfUkf, UkfA e UkfO é mostrada na Figura 9.1. Notar que o gráfico de PfUkf só vai até a iteração 8. Isso se deve ao fato de o número de iterações do Algoritmo 3

ser $\frac{budget}{N_s}$, mas vale lembrar que o algoritmo efetua N_s avaliações da função objetivo a cada iteração. Os parâmetros encontrados pelos processos de sintonia são mostrados na Tabela 9.1.

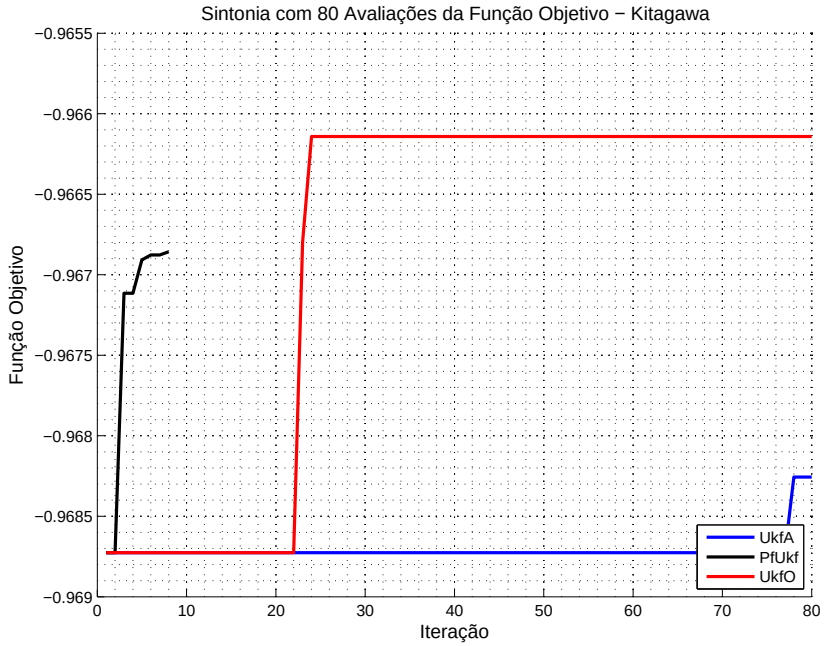

Figura 9.1: *Kitagawa - Sintonia dos Filtros - Ruídos Conhecidos*

Tabela 9.1: *Kitagawa - Parâmetros da UT - Ruídos Conhecidos*

	α	β	κ
UkfO	0.31929	1.99620	1.71978
UkfA	0.35392	1.46297	4.01379
PfUkf	0.82318	2.16289	0.02967

9.1.4 Estimativas e Credibilidade

Os erros de estimação de estado são mostrados na Tabela 9.2. A primeira linha da tabela mostra o erro médio quadrático das estimativas de estado (MSEx) produzidas por cada um dos filtros nas séries temporais de teste. A segunda linha mostra o valor absoluto da variação que caracteriza o intervalo de confiança de 95% do MSEx de cada um dos filtros.

A terceira e quarta linhas, que seguem a mesma logica das duas primeiras linhas, apresentam os resultados segundo o critério NCI, enquanto a quinta linha mostra o tempo médio de execução dos filtros. A ideia da quinta linha é evidenciar o fato de que o processo de sintonia *offline* não afeta os tempos de execução do UKF e, além disso, poder avaliar empiricamente o custo computacional da abordagem de treinamento *online* utilizada pelo Ukf-ML.

Tabela 9.2: *Kitagawa - Erros nas Estimativas de Estado - Ruídos Conhecidos*

	UkfD	CKF	UkfO	Ukf-ML	EKF	UkfA	PfUkf
MSEx	0.720	10.466	0.181	1.385	34.964	0.146	0.142
$\Delta_{.95X}$	0.153	3.955	0.092	0.344	18.414	0.088	0.084
NCI_{μ}	10.822	53.457	8.482	21.487	58.596	9.076	8.967
$\Delta_{.95NCI}$	11.809	4.942	11.032	11.394	4.444	11.191	11.153
Tempo Médio (s)	0.003	0.003	0.003	0.054	0.001	0.003	0.003

Tabela 9.3: *Kitagawa - Erros nas Estimativas de Medida - Ruídos Conhecidos*

	UkfD	CKF	UkfO	Ukf-ML	EKF	UkfA	PfUkf
MSEy	5.098	5.849	3.965	5.813	8.997	3.290	3.255
$\Delta_{.95Y}$	0.787	0.920	0.700	0.909	1.304	0.550	0.527
NLLy	3.069	42.818	2.509	2.910	775.116	1.812	1.716

A distribuição dos MSEx e a evolução dos NCI calculados em cada uma das séries de teste são mostradas na figuras 9.2 e 9.3.

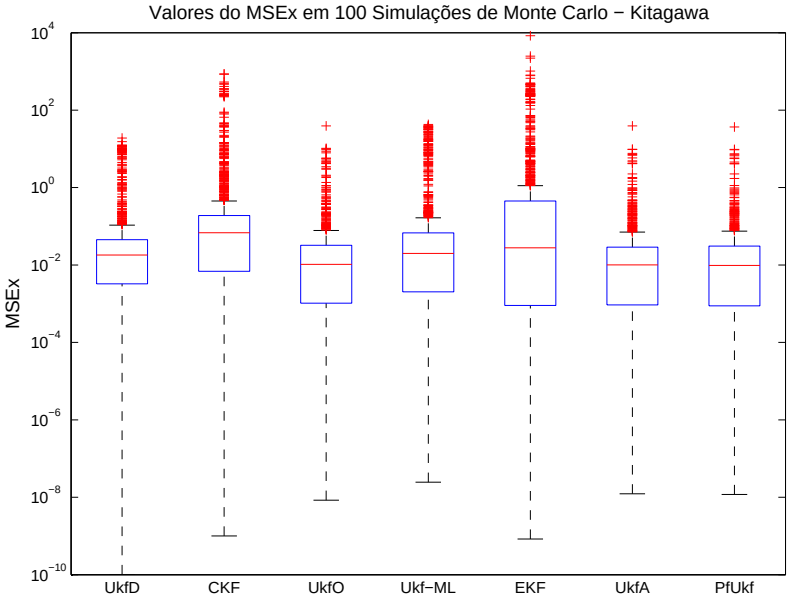

Figura 9.2: *Kitagawa - Erros nas Estimativas de Estado - Ruídos Conhecidos*

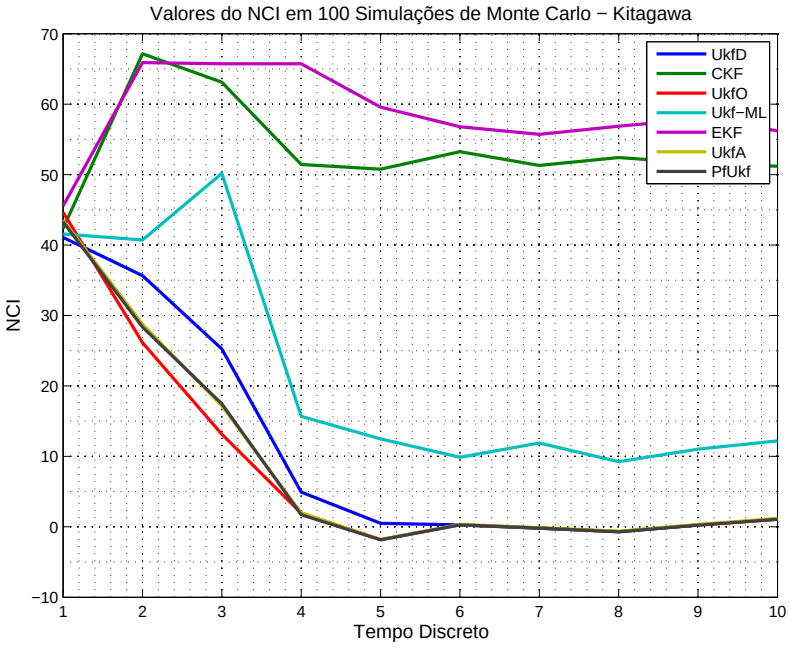

Figura 9.3: *Kitagawa - Índices de Não-Credibilidade - Ruídos Conhecidos*

9.1.5 Análise

Considerando a Figura 9.1, vemos que os processos de otimização chegaram a valores muito similares da função objetivo. Apesar disso, a Tabela 9.1 mostra que as sintonias obtidas são bastante diferentes.

A Tabela 9.4 resume os resultados do experimento, de acordo com cada critério adotado. O critério mais importante é o MSEx, seguido do NCI, pois estamos interessados na capacidade dos filtros de prever o estado do sistema dinâmico.

Tabela 9.4: *Kitagawa - Classificação dos Filtros - Ruídos Conhecidos*

	1	2	3	4	5	6	7
MSEx	PfUkf	UkfA	UkfO	UkfD	Ukf-ML	CKF	EKF
NCI_μ	UkfO	PfUkf	UkfA	UkfD	Ukf-ML	CKF	EKF
MSEy	PfUkf	UkfA	UkfO	UkfD	Ukf-ML	CKF	EKF
NLLy	PfUkf	UkfA	UkfO	Ukf-ML	UkfD	CKF	EKF

A Tabela 9.1 mostra que os filtros sintonizados *offline* tiveram desempenhos muito superiores aos demais filtros. Considerando o critério MSEx, o UkfA e o PfUkf foram melhores que o UkfO. O Ukf-ML, que utiliza sintonia *online*, apresentou desempenho surpreendentemente inferior, pior até mesmo que o do UkfD.

Os desempenhos bastante insatisfatórios do EKF e do CKF são refletidos na Figura 9.2 pela presença de quantidade significativa de grandes valores de MSEx, indicando que esses filtros tiverem desempenhos particularmente ruins em muitas séries de teste.

Quanto à credibilidade, a Figura 9.3 mostra que os filtros UkfD, UkfA, UkfO e PfUkf apresentaram índices similares, enquanto os demais, incluindo Ukf-ML, foram bem menos confiáveis.

É interessante observar que a classificação dos filtros segundo o critério MSEy coincidiu com a classificação segundo o critério MSEx. O critério NLLy só não classificou corretamente

os filtros UkfD e Ukf-ML.

9.2 Sinusoid

9.2.1 Dados de Treinamento

O processo de otimização realizado pelos algoritmos UkfA, UkfO e PfUkf fez uso de uma única série temporal composta por 1000 observações sequenciais. A distribuição do estado inicial usada foi $x_0 \sim \mathcal{N}(0, 1)$.

9.2.2 Dados de Teste

Para avaliação de desempenho, todos os filtros foram aplicados a 100 séries temporais independentes, de 100 medidas cada. A distribuição do estado inicial usada foi $x_0 \sim \mathcal{N}(0, 1)$.

9.2.3 Sintonia

A sintonia do UkfA usou o critério EI (7.4). A evolução do valor máximo da função objetivo em cada iteração do processo de otimização é apresentada na figura 9.4. Os parâmetros da UT obtidos são mostrados na Tabela 9.5.

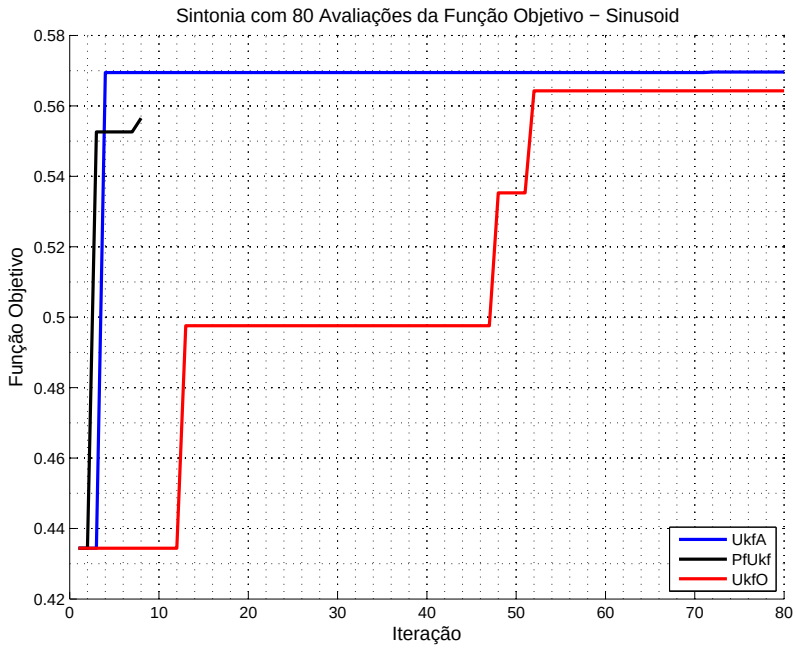

Figura 9.4: *Sinusoid - Sintonia dos filtros - Ruídos Conhecidos*

Tabela 9.5: *Sinusoid - Parâmetros da UT - Ruídos Conhecidos*

	α	β	κ
UkfO	2.03333	0.24597	0.37982
UkfA	2.31416	1.52518	0.00934
PfUkf	2.54021	2.13632	0.00000

9.2.4 Estimativas e Credibilidade

As estatísticas das estimativas de estado e de medidas são respectivamente apresentadas nas Tabelas 9.6 e 9.7.

Tabela 9.6: *Sinusoid - Erros nas Estimativas de Estado - Ruídos Conhecidos*

	UkfD	CKF	UkfO	Ukf-ML	EKF	UkfA	PfUkf
MSEx	0.966	1.449	0.761	1.416	1.515	0.760	0.791
$\Delta_{.95X}$	0.027	0.059	0.023	0.052	0.057	0.024	0.025
NCI_{μ}	-0.735	6.716	0.293	2.598	3.135	0.405	0.632
$\Delta_{.95NCI}$	0.128	0.328	0.147	0.214	0.236	0.154	0.144
Tempo Médio (s)	0.029	0.029	0.029	0.530	0.006	0.029	0.029

Tabela 9.7: *Sinusoid - Erros nas Estimativas de Medida - Ruídos Conhecidos*

	UkfD	CKF	UkfO	Ukf-ML	EKF	UkfA	PfUkf
MSEy	0.023	0.023	0.018	0.025	0.030	0.018	0.019
$\Delta_{.95Y}$	0.001	0.001	0.001	0.001	0.001	0.001	0.001
NLLy	-0.465	-0.345	-0.575	-0.356	-0.192	-0.575	-0.563

A distribuição dos erros quadráticos de estimação de estado nas séries de teste é ilustrada na Figura 9.5, enquanto a evolução do NCI dos filtros nos testes é apresentado na Figura 9.6.

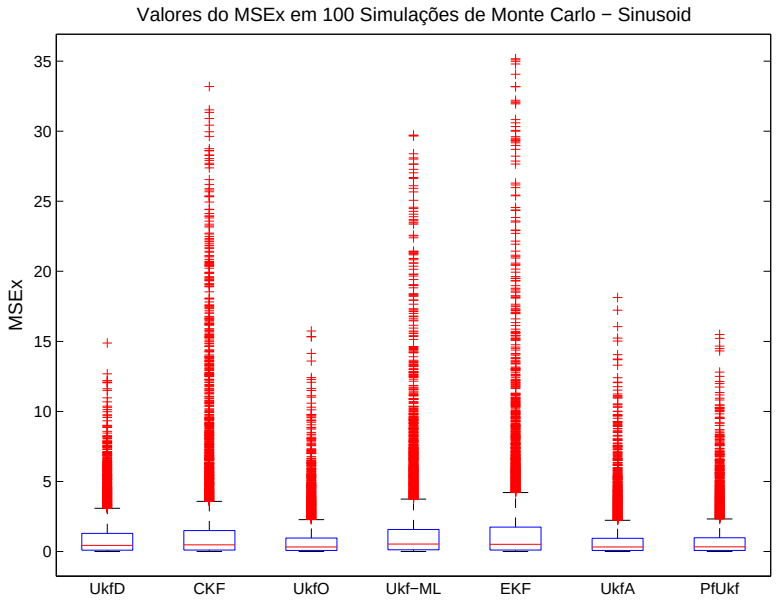

Figura 9.5: *Sinusoid - Erros nas Estimativas de Estado - Ruídos Conhecidos*

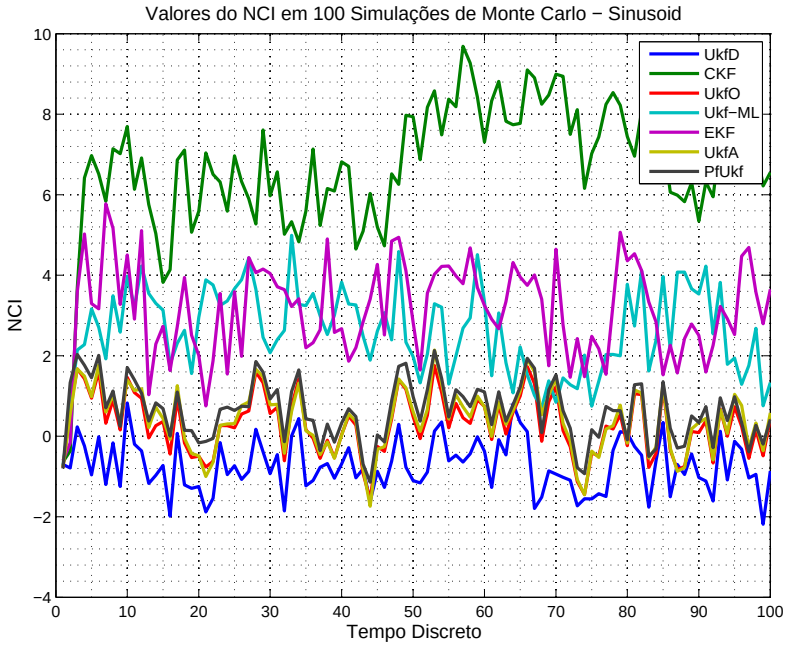

Figura 9.6: *Sinusoid - Índices de Não-Credibilidade - Ruídos Conhecidos*

9.2.5 Análise

Tabela 9.8: *Sinusoid - Classificação dos Filtros - Ruídos Conhecidos*

	1	2	3	4	5	6	7
MSEx	UkfA	UkfO	PfUkf	UkfD	Ukf-ML	CKF	EKF
NCI_μ	UkfO	UkfA	PfUkf	UkfD	Ukf-ML	EKF	CKF
MSEy	UkfA	UkfO	PfUkf	UkfD	CKF	Ukf-ML	EKF
NLLy	UkfO	UkfA	PfUkf	UkfD	Ukf-ML	CKF	EKF

A Figura 9.4 mostra que novamente os processos de sintonia do UkfA, UkfO e PfUkf chegaram a valores próximos da função objetivo, embora as sintonias obtidas sejam diferentes.

Em termos de MSEx, o UkfA e o UkfO foram os dois melhores filtros, seguidos de perto pelo PfUkf. Como em Kitagawa, o Ukf-ML apresentou mau desempenho, não justificando o custo computacional extra que impõe.

A Figura 9.5 mostra que mesmo os três melhores filtros apresentaram um bom número de *outliers*, indicando que Sinusoid é um problema de filtragem interessante para testar filtros não-lineares.

Com relação às medidas, nem o critério NLLy nem o critério MSEy concordaram plenamente com o critério MSEx na classificação dos filtros. Os erros, porém, ocorreram na classificação de filtros com valores MSEx parecidos.

Os filtros UkfO, UkfA, PfUkf e UkfD foram os mais críveis. Os demais filtros apresentaram credibilidades bem inferiores.

9.3 Bearings Only

9.3.1 Dados de Treinamento

O processo de otimização dos algoritmos UkfA, UkfO e PfUkf fez uso de cinco séries temporais compostas cada uma por 1000 observações sequenciais. A distribuição do estado inicial usado para a geração dos dados de treinamento foi

$$\mathbf{x}_0 \sim \mathcal{N}\left(\begin{bmatrix}20\\5\end{bmatrix}, \begin{bmatrix}0.1 & 0.01\\0.01 & 0.1\end{bmatrix}\right) . \tag{9.1}$$

9.3.2 Dados de Teste

Para avaliação de desempenho, todos os filtros foram aplicados a 100 séries temporais independentes, de 100 medidas cada. A distribuição do estado inicial usado para a geração dos dados de teste foi a mesma usada para a geração dos dados de treinamento.

9.3.3 Sintonia

A sintonia do UkfA usou o critério EI (7.4). A evolução do valor máximo da função objetivo em cada iteração do processo de otimização é apresentada na Figura 9.7. Os parâmetros resultantes são apresentados na Tabela 9.9.

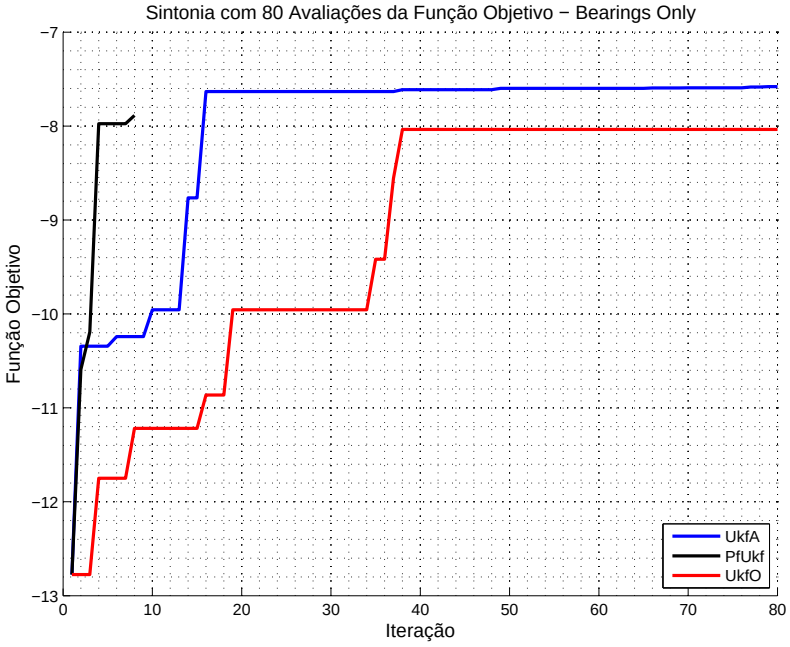

Figura 9.7: *Bearings Only - Sintonia dos Filtros - Ruídos Conhecidos*

Tabela 9.9: *Bearings Only - Parâmetros da UT - Ruídos Conhecidos*

	α	β	κ
UkfO	0.95613	1.64537	4.81533
UkfA	1.16809	3.99824	3.51218
PfUkf	1.24792	3.77163	2.05941

9.3.4 Estimativas e Credibilidade

As estatísticas das estimativas de estado podem ser vistas na Tabela 9.10 e na Figura 9.8.

Tabela 9.10: *Bearings Only - Erros nas Estimativas de Estado - Ruídos Conhecidos*

	UkfD	CKF	UkfO	Ukf-ML	EKF	UkfA	PfUkf
MSEx	9.208	16.569	3.454	1.532	52.056	2.563	2.846
$\Delta_{.95X}$	0.556	0.794	0.308	0.045	1.200	0.171	0.253
NCI_{μ}	7.731	9.596	3.635	1.091	13.878	2.236	2.443
$\Delta_{.95NCI}$	1.020	1.251	0.478	0.186	1.630	0.337	0.412
Tempo Médio (s)	0.032	0.032	0.032	0.602	0.006	0.032	0.032

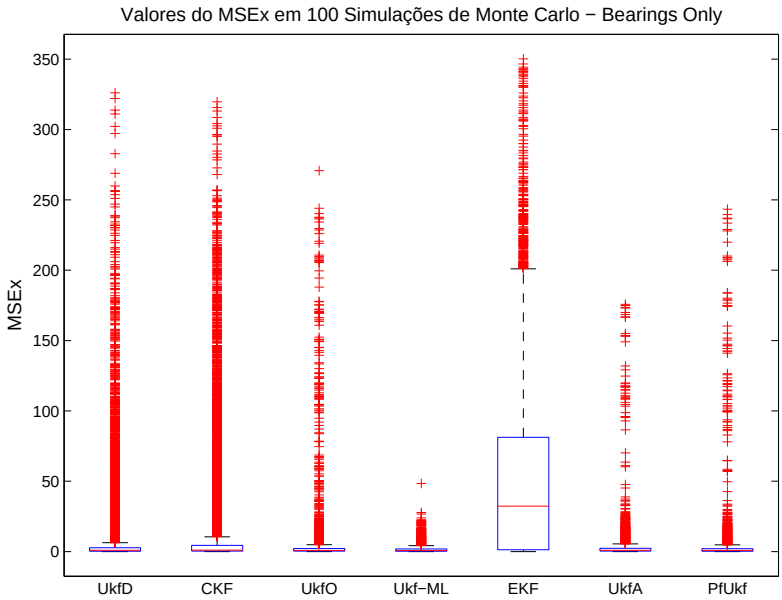

Figura 9.8: *Bearings Only - Erros nas Estimativas de Estado - Ruídos Conhecidos*

Tabela 9.11: *Bearings Only - Erros nas Estimativas de Medida - Ruídos Conhecidos*

	UkfD	CKF	UkfO	Ukf-ML	EKF	UkfA	PfUkf
MSEy	0.905	0.957	1.002	0.577	1.528	1.038	0.959
$\Delta_{.95Y}$	0.039	0.036	0.047	0.030	0.046	0.049	0.047
NLLy	3.718	6.369	1.405	0.346	20.984	1.184	1.136

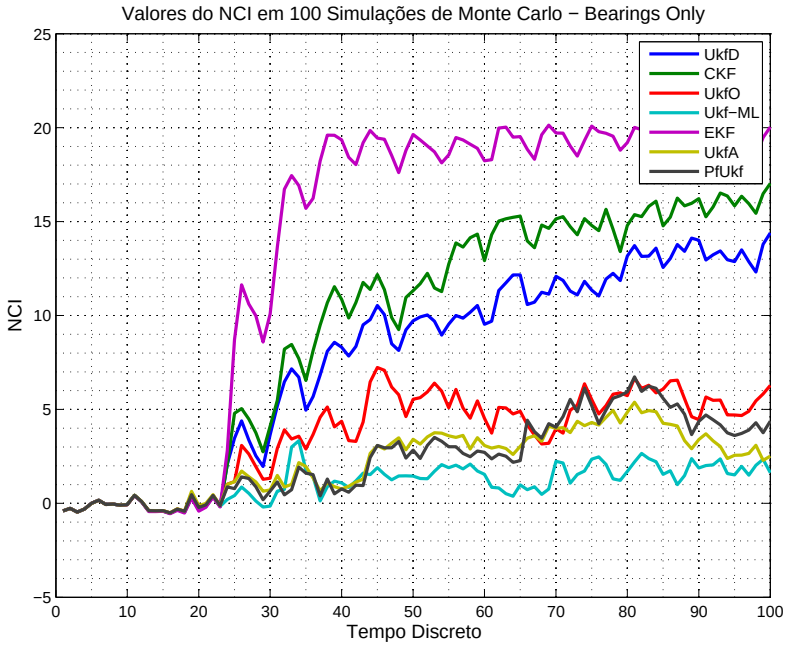

Figura 9.9: *Bearings Only - Índices de Não-Credibilidade - Ruídos Conhecidos*

9.3.5 Análise

Tabela 9.12: *Bearings Only - Classificação dos Filtros - Ruídos Conhecidos*

	1	2	3	4	5	6	7
MSEx	Ukf-ML	UkfA	PfUkf	UkfO	UkfD	CKF	EKF
NCI_μ	Ukf-ML	UkfA	PfUkf	UkfO	UkfD	CKF	EKF
MSEy	Ukf-ML	UkfD	CKF	PfUkf	UkfO	UkfA	EKF
NLLy	Ukf-ML	PfUkf	UkfA	UkfO	UkfD	CKF	EKF

A Figura 9.7 mostra que os processos de sintonia atingiram valores da função objetivo relativamente próximos, embora com sintonias diferentes.

Em termos do critério MSEx, o melhor filtro foi o Ukf-ML, seguido do UkfA, do PfUkf e, um pouco mais distante, do UkfO. Os demais filtros apresentaram desempenhos bastante inferiores. Em concordância com essa classificação, o filtro mais crível foi o Ukf-ML, seguido do UkfA, do PfUkf e do UkfO. Os demais filtros apresentaram credibilidades muito inferiores.

Em termos de estimativa de medidas, o critério NLLy classificou os filtros de modo mais condizente com o critério MSEx que o critério MSEy.

Capítulo 10

Sintonia com Identificação Conjunta das Covariâncias dos Ruídos

Além dos parâmetros que determinam o funcionamento da UT, o UKF utiliza mais dois conjuntos de parâmetros, que são os componentes das matrizes de covariância dos ruídos de processo e de medida. Nos experimentos descritos no Capítulo 9, somente os parâmetros da UT são sintonizados. Haverá situações, porém, nas quais será necessário determinar valores para as covariâncias dos ruídos de processo e de medida.

Assim, neste capítulo, são realizados experimentos que envolvem a sintonia simultânea dos parâmetros da UT e dos elementos diagonais das matrizes de covariância de ruído usadas no filtro. Os experimentos adotam os Algoritmos 3 e 5, também utilizados no Capítulo 9, com a diferença de que, aqui, o espaço de parâmetros é formado pelos três parâmetros da UT e pelos elementos das diagonais principais das matrizes dos ruídos de processo e de medida. Para os Algoritmos 3 e 5, os parâmetros livres são dados na Tabela 10.1.

Tabela 10.1: *Parâmetros Livres dos Algoritmos 3 e 5*

Parâmetro	**Mínimo**	**Máximo**
α	0.01	4
β	0	4
κ	0	5
Qf_{ii}	$0.25Q_{ii}$	$1.75Q_{ii}$
Rf_{ii}	$0.25R_{ii}$	$1.75R_{ii}$
Parâmetro	**Valor**	
N_s	10	
budget	80	

Na Tabela 10.1, Q_{ii} e R_{ii} correspondem respectivamente aos elementos ii das diagonais principais das matrizes de covariância dos ruídos de processo e de medida usadas na geração dos dados de treinamento e de teste, enquanto Qf_{ii} e Rf_{ii} são os elementos correspondentes das matrizes de covariância de ruídos usadas pelos filtros. Essas alterações em relação ao Capítulo 9 dão origem ao UkfA que inclui a sintonia das matrizes de covariância dos ruídos de processo e de medida (UkfA-QR), ao PfUkf que inclui a sintonia das matrizes de covariância dos ruídos de processo e de medida (PfUkf-QR) e ao UkfO que inclui a sintonia das matrizes de covariância dos ruídos de processo e de medida (UkfO-QR).

Para tornar os experimentos mais interessantes, os filtros UkfD, CKF, EKF e Ukf-ML foram executados com os valores reais das matrizes de covariância dos ruídos de processo e de medida.

10.1 Kitagawa

10.1.1 Dados de Treinamento

O processo de otimização fez uso de 10 séries temporais de 1000 medidas sequenciais. O conjunto de pontos iniciais $\mathbf{\Theta}$ foi composto de 32 amostras geradas como grade de pontos no hipercubo definido pela Tabela 10.1. A distribuição do estado inicial usada para gerar a série de treinamento foi $x_0 \sim \mathcal{N}(0, 0.5^2)$.

10.1.2 Dados de Teste

Os dados de teste consistiram de 100 séries temporais de 10 medidas cada. A distribuição do estado inicial usada para gerar as séries de teste também foi $x_0 \sim \mathcal{N}(0, 0.5^2)$.

10.1.3 Sintonia

O processo de sintonia do UkfA-QR usou o critério EI (7.4). A evolução da sintonia dos filtros PfUkf-QR, UkfA-QR e UkfO-QR é mostrada na Figura 10.1. Os parâmetros encontrados pelos processos de sintonia são mostrados na Tabela 10.2.

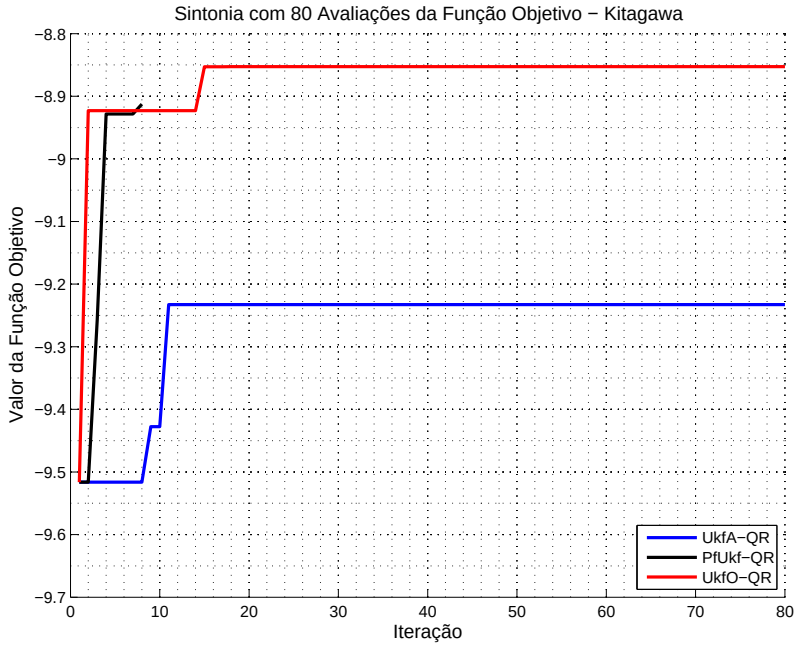

Figura 10.1: *Kitagawa - Sintonia dos Filtros - Ruídos Identificados*

Tabela 10.2: *Kitagawa - Parâmetros da UT - Ruídos Identificados*

	α	β	κ	Q_{11}	R_{11}
UkfO-QR	0.50987	1.95642	0.55883	0.03758	0.00006
UkfA-QR	0.59954	1.96899	0.64568	0.04522	0.00017
PfUkf-QR	1.06125	2.35973	0.05061	0.03637	0.00007

10.1.4 Estimativas e Credibilidade

Os erros de estimação de estado são mostrados na Tabela 10.3.

Tabela 10.3: *Kitagawa - Erros nas Estimativas de Estado - Ruídos Identificados*

	UkfD	CKF	UkfO-QR	Ukf-ML	EKF	UkfA-QR	PfUkf-QR
MSEx	0.720	10.466	0.162	1.385	34.964	0.147	0.160
$\Delta_{.95X}$	0.153	3.955	0.091	0.344	18.414	0.087	0.070
NCI_{μ}	10.822	53.457	9.143	21.487	58.596	8.037	9.395
$\Delta_{.95NCI}$	11.809	4.942	11.363	11.394	4.444	10.798	11.053
Tempo Médio (s)	0.004	0.004	0.004	0.060	0.001	0.004	0.004

Tabela 10.4: *Kitagawa - Erros nas Estimativas de Medida - Ruídos Identificados*

	UkfD	CKF	UkfO-QR	Ukf-ML	EKF	UkfA-QR	PfUkf-QR
MSEy	5.098	5.849	3.689	5.813	8.997	3.381	3.801
$\Delta_{.95Y}$	0.787	0.920	0.649	0.909	1.304	0.570	0.587
NLLy	3.069	42.818	2.283	2.910	775.116	1.747	2.092

A distribuição dos MSEx e a evolução dos NCI calculados em cada uma das séries de teste são mostradas na figuras 10.2 e 10.3.

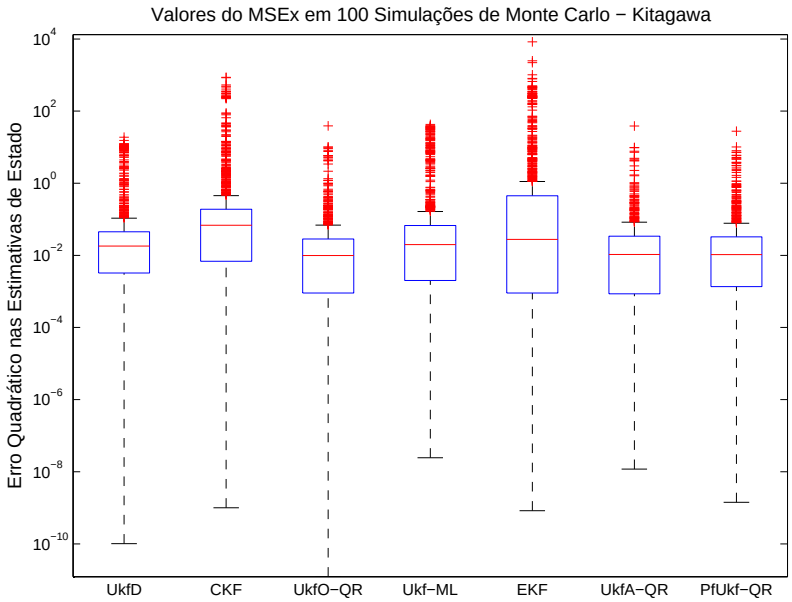

Figura 10.2: *Kitagawa - Erros nas Estimativas de Estado - Ruídos Identificados*

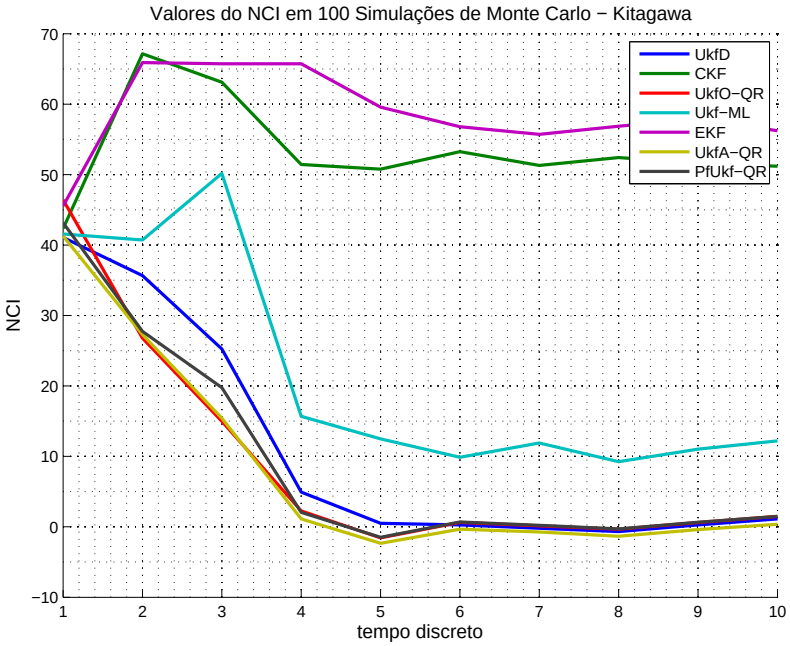

Figura 10.3: *Kitagawa - Índices de Não-Credibilidade - Ruídos Identificados*

10.1.5 Análise

Tabela 10.5: *Kitagawa - Classificação dos Filtros - Ruídos Identificados*

	1	2	3	4	5	6	7
MSEx	UkfA-QR	PfUkf-QR	UkfO-QR	UkfD	Ukf-ML	CKF	EKF
NCI_μ	UkfA-QR	UkfO-QR	PfUkf-QR	UkfD	Ukf-ML	CKF	EKF
MSEy	UkfA-QR	UkfO-QR	PfUkf-QR	UkfD	Ukf-ML	CKF	EKF
NLLy	UkfA-QR	PfUkf-QR	UkfO-QR	Ukf-ML	UkfD	CKF	EKF

Considerando o critério MSEx, UkfA-QR foi o melhor dos filtros, seguido pelo PfUkf-QR e pelo UkfO-QR. O Ukf-ML teve novamente desempenho pior que o do UkfD. Os desempenhos bastante insatisfatórios do CKF e do EKF são refletidos na Figura 10.2 pela presença de quantidade significativa de grandes valores de MSEx, indicando que esses filtros tiverem desempenhos particularmente ruins em muitas séries de teste.

Quanto à concordância entre o desempenho nas estimativas de medidas e o desempenho nas estimativas de estado, os critérios NLLy e MSEy seriam boas referências, pois foram capazes de identificar os 3 melhores filtros e os dois piores.

Quanto à credibilidade, os filtros UkfD, UkfA-QR, UkfO-QR e PfUkf-QR apresentaram índices similares, enquanto os demais, incluindo Ukf-ML, foram bem menos confiáveis.

10.2 Sinusoid

10.2.1 Dados de Treinamento

O processo de otimização realizado pelos algoritmos UkfA-QR, UkfO-QR e PfUkf-QR fez uso de cinco séries temporais compostas por 1000 observações sequenciais. O conjunto de

pontos iniciais $\mathbf{\Theta}$ foi composto de 32 amostras geradas como grade de pontos no hipercubo definido pela Tabela 10.1. A distribuição do estado inicial usada foi $x_0 \sim \mathcal{N}(0, 1)$.

10.2.2 Dados de Teste

Para avaliação de desempenho, todos os filtros foram aplicados a 100 séries temporais independentes, de 100 medidas cada. A distribuição do estado inicial usada foi $x_0 \sim \mathcal{N}(0, 1)$.

10.2.3 Sintonia

A sintonia do UkfA-QR usou o critério EI (7.4). A evolução do valor máximo da função objetivo em cada iteração do processo de otimização é apresentada na figura 10.4. Os parâmetros da UT obtidos são mostrados na Tabela 10.6.

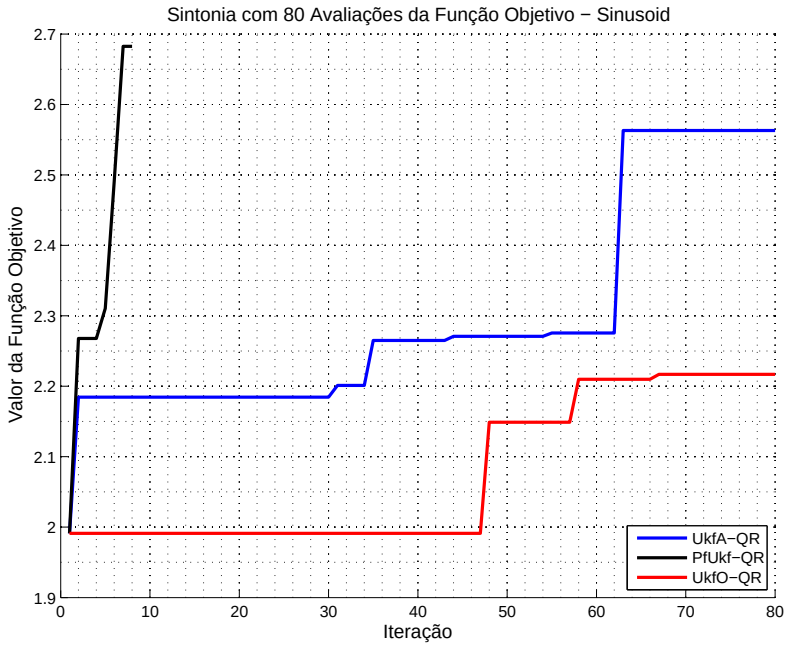

Figura 10.4: *Sinusoid - Sintonia dos Filtros - Ruídos Identificados*

Tabela 10.6: *Sinusoid - Parâmetros da UT - Ruídos Identificados*

	α	β	κ	Q_{11}	R_{11}
UkfO-QR	1.43468	1.08539	0.45042	0.01300	0.00925
UkfA-QR	1.91544	0.18379	0.34880	0.01749	0.00821
PfUkf-QR	2.20819	2.19856	0.00000	0.00262	0.01235

10.2.4 Estimativas e Credibilidade

As estatísticas das estimativas de estado e de medidas são respectivamente apresentadas nas Tabelas 10.7 e 10.8.

Tabela 10.7: *Sinusoid - Erros nas Estimativas de Estado - Ruídos Identificados*

	UkfD	CKF	UkfO-QR	Ukf-ML	EKF	UkfA-QR	PfUkf-QR
MSEx	0.966	1.449	0.973	1.416	1.515	0.783	0.831
$\Delta_{.95X}$	0.027	0.059	0.027	0.052	0.057	0.025	0.023
NCI_{μ}	-0.735	6.716	-0.454	2.598	3.135	0.791	-0.726
$\Delta_{.95NCI}$	0.128	0.328	0.129	0.214	0.236	0.162	0.135
Tempo Médio (s)	0.030	0.030	0.030	0.537	0.006	0.030	0.030

Tabela 10.8: *Sinusoid - Erros nas Estimativas de Medida - Ruídos Identificados*

	UkfD	CKF	UkfO-QR	Ukf-ML	EKF	UkfA-QR	PfUkf-QR
MSEy	0.023	0.023	0.023	0.025	0.030	0.019	0.019
$\Delta_{.95Y}$	0.001	0.001	0.001	0.001	0.001	0.001	0.001
NLLy	-0.465	-0.345	-0.465	-0.356	-0.192	-0.539	-0.553

A distribuição dos erros quadráticos de estimação de estado nas séries de teste é ilustrada na Figura 10.5, enquanto a evolução do NCI dos filtros nos testes é apresentado na Figura 10.6.

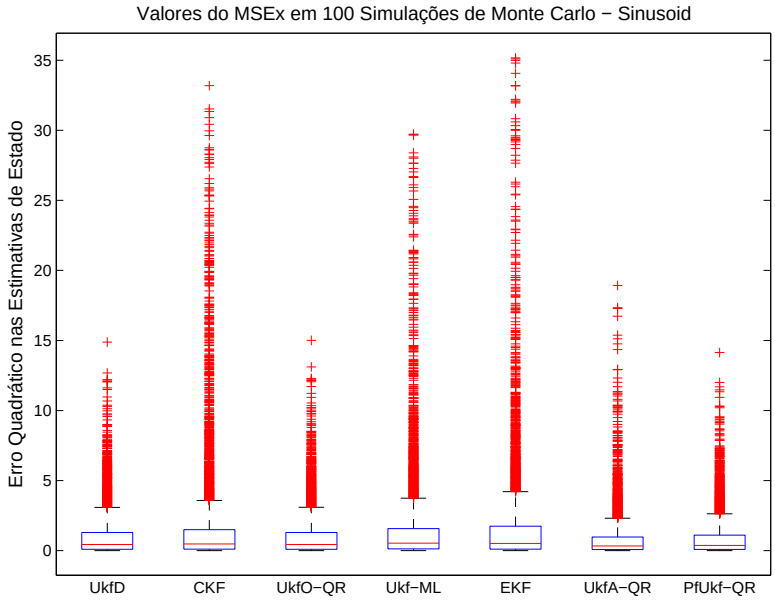

Figura 10.5: *Sinusoid - Erros nas Estimativas de Estado - Ruídos Identificados*

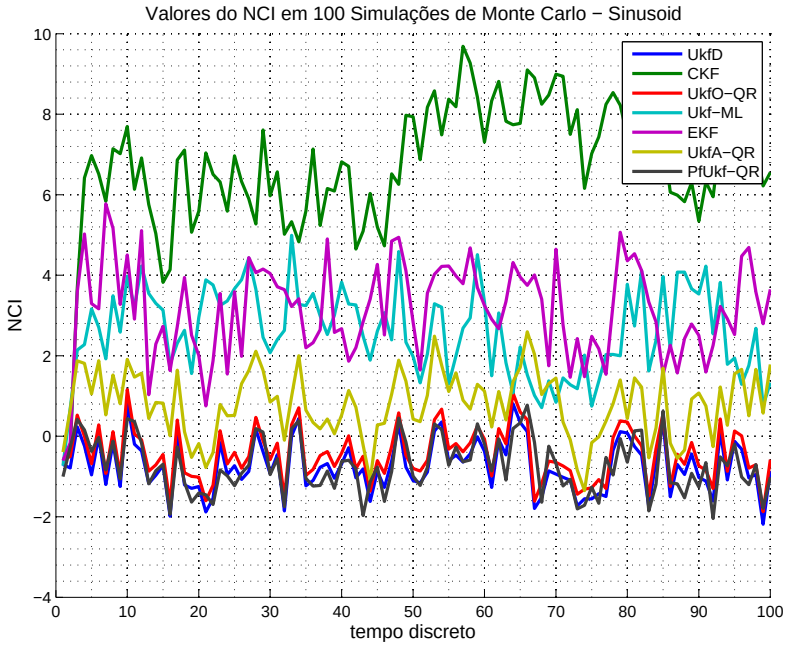

Figura 10.6: *Sinusoid - Índices de Não-Credibilidade - Ruídos Identificados*

10.2.5 Análise

Tabela 10.9: *Sinusoid - Classificação dos Filtros - Ruídos Identificados*

	1	2	3	4	5	6	7
MSEx	UkfA-QR	PfUkf-QR	UkfD	UkfO-QR	Ukf-ML	CKF	EKF
NCI_μ	UkfO-QR	PfUkf-QR	UkfD	UkfA-QR	Ukf-ML	EKF	CKF
MSEy	PfUkf-QR	UkfA-QR	UkfD	UkfO-QR	CKF	Ukf-ML	EKF
NLLy	PfUkf-QR	UkfA-QR	UkfO-QR	UkfD	Ukf-ML	CKF	EKF

Em termos de MSEx, os filtros PfUkf-QR e UkfA-QR foram melhores que o UkfO-QR, que ficou praticamente empatado com o UkfD. Como em Kitagawa, o Ukf-ML apresentou mau desempenho, não justificando o custo computacional extra que impõe.

Com relação às medidas, os critério NLLy e MSEy foram capazes de identificar os dois melhores e o pior filtro.

Quanto à confiabilidade, os filtros PfUkf-QR,UkfA-QR, UkfD e UkfO-QR apresentaram índices de credibilidade bem superiores aos dos demais filtros.

10.3 Bearings Only

10.3.1 Dados de Treinamento

O processo de sintonia dos filtros UkfA-QR, UkfO-QR e PfUkf-QR fez uso de 5 séries temporais compostas cada uma por 1000 observações sequenciais. O conjunto de pontos iniciais $\mathbf{\Theta}$ foi composto de 64 amostras geradas como grade de pontos no hipercubo definido pela Tabela 10.1. A distribuição do estado inicial usado para a geração dos dados de treinamento foi

$$\mathbf{x}_0 \sim \mathcal{N}\left(\begin{bmatrix}20\\5\end{bmatrix}, \begin{bmatrix}0.1 & 0.01\\0.01 & 0.1\end{bmatrix}\right) . \tag{10.1}$$

10.3.2 Dados de Teste

Para avaliação de desempenho, todos os filtros foram aplicados a 100 séries temporais independentes, de 100 medidas cada. A distribuição do estado inicial usado para a geração dos dados de teste foi a mesma usada para a geração dos dados de treinamento.

10.3.3 Sintonia

A sintonia do UkfA-QR fez uso do critério EI (7.4). A evolução do valor máximo da função objetivo em cada iteração do processo de otimização é apresentada na Figura 10.7. Os parâmetros resultantes são apresentados na Tabela 10.10.

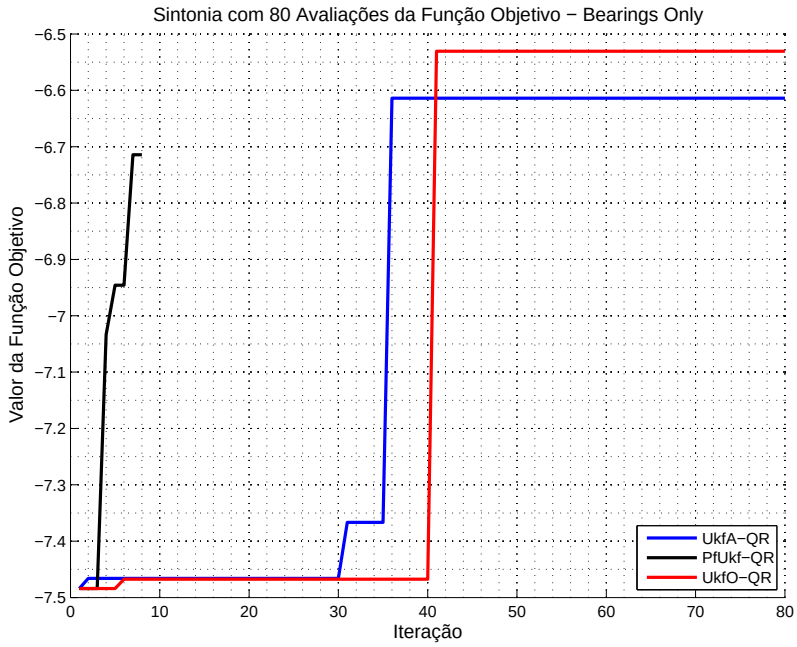

Figura 10.7: *Bearings Only - Sintonia dos Filtros - Ruídos Identificados*

Tabela 10.10: *Bearings Only - Parâmetros da UT - Ruídos Identificados*

	α	β	κ	Q_{11}	Q_{22}	R_{11}
UkfO-QR	1.38953	3.36558	0.57605	0.11548	0.13108	0.03646
UkfA-QR	0.84558	3.98514	3.34696	0.17154	0.08923	0.03719
PfUkf-QR	0.98765	2.46818	3.82794	0.10162	0.12198	0.04079

10.3.4 Estimativas e Credibilidade

As estatísticas das estimativas de estado podem ser vistas na Figura 10.8 e na Tabela 10.11.

Tabela 10.11: *Bearings Only - Erros nas Estimativas de Estado - Ruídos Identificados*

	UkfD	CKF	UkfO-QR	Ukf-ML	EKF	UkfA-QR	PfUkf-QR
MSEx	9.208	16.569	3.153	1.532	52.056	1.971	2.422
$\Delta_{.95X}$	0.556	0.794	0.277	0.045	1.200	0.071	0.150
NCI_{μ}	7.731	9.596	2.287	1.091	13.878	-0.183	1.825
$\Delta_{.95NCI}$	1.020	1.251	0.482	0.186	1.630	0.188	0.373
Tempo Médio (s)	0.032	0.032	0.032	0.598	0.006	0.032	0.032

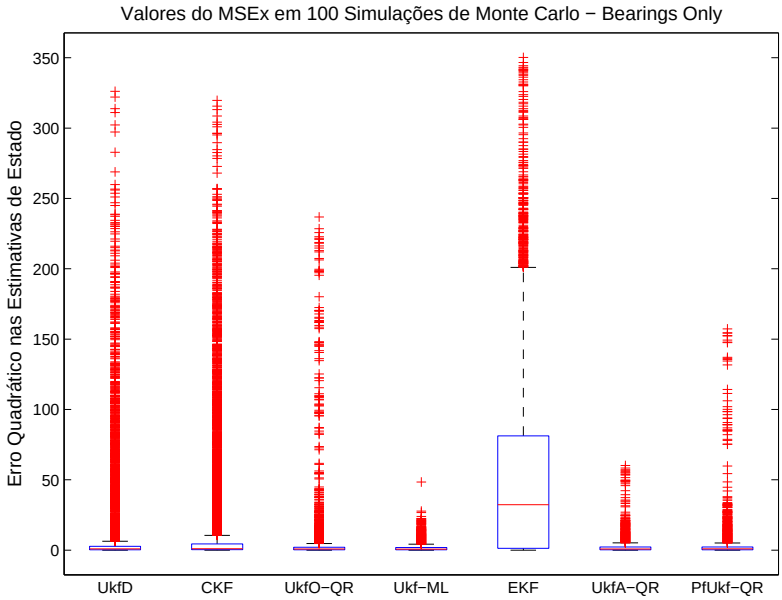

Figura 10.8: *Bearings Only - Erros nas Estimativas de Estado - Ruídos Identificados*

Tabela 10.12: *Bearings Only - Erros nas Estimativas de Medida - Ruídos Identificados*

	UkfD	CKF	UkfO-QR	Ukf-ML	EKF	UkfA-QR	PfUkf-QR
MSEy	0.905	0.957	0.935	0.577	1.528	0.869	0.938
$\Delta_{.95Y}$	0.039	0.036	0.044	0.030	0.046	0.040	0.045
NLLy	3.718	6.369	1.139	0.346	20.984	0.981	1.029

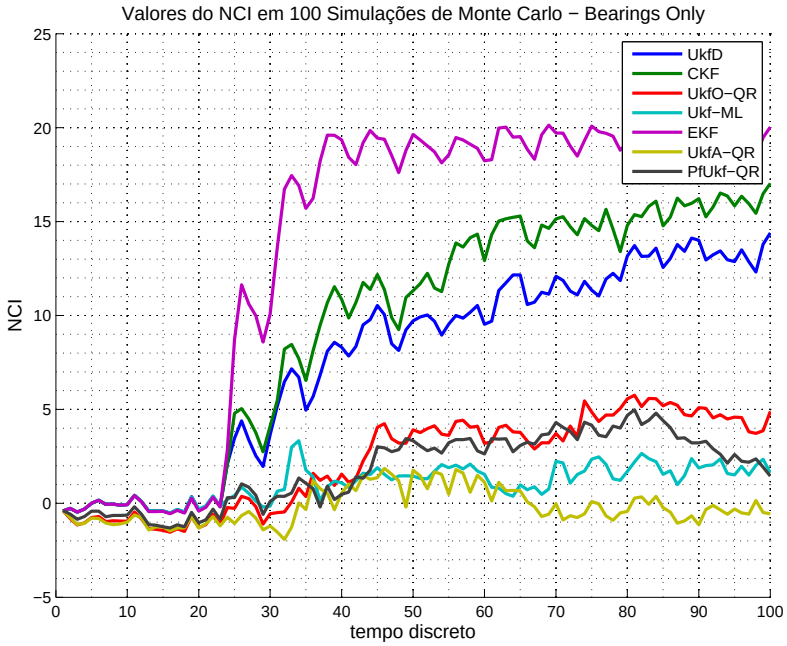

Figura 10.9: *Bearings Only - Índices de Não-Credibilidade - Ruídos Identificados*

10.3.5 Análise

Tabela 10.13: *Bearings Only - Classificação dos Filtros - Ruídos Identificados*

	1	2	3	4	5	6	7
MSEx	Ukf-ML	UkfA-QR	PfUkf-QR	UkfO-QR	UkfD	CKF	EKF
NCI_μ	UkfA-QR	Ukf-ML	PfUkf-QR	UkfO-QR	UkfD	CKF	EKF
MSEy	Ukf-ML	UkfA-QR	UkfD	UkfO-QR	PfUkf-QR	CKF	EKF
NLLy	Ukf-ML	UkfA-QR	PfUkf-QR	UkfO-QR	UkfD	CKF	EKF

Em termos de MSEx, o melhor filtro foi, finalmente, o Ukf-ML, seguido do UkfA-QR, do PfUkf-QR e do UkfO-QR. Os demais filtros apresentaram desempenhos bastante inferiores. Em concordância com essa classificação, o filtros mais críveis foram o Ukf-ML, o UkfA-QR e o PfUkf-QR, seguidos do UkfO-QR. Os demais filtros apresentaram credibilidades muito inferiores.

Os critérios NLLy e MSEy conseguem prever os dois melhores e os dois piores filtros em termos de MSEx.

Capítulo 11

Múltiplos Estados Iniciais

Uma importante questão pertinente à estratégia de sintonia utilizada nos Algoritmos 3 e 5 é se a sintonia teria de ser refeita todas as vezes que o filtro recebesse uma nova distribuição do estado inicial (a forma da distribuição continuaria sendo Gaussiana, mas as médias e/ou covariâncias seriam diferentes). Se a resposta a essa pergunta fosse sim, então esses algoritmos teriam de ser executados *online*, o que os tornaria bem menos interessantes.

Para responder a essa questão, vamos primeiramente analisar a concepção da função objetivo definida no Algoritmo 1. Ela não impõe restrições à maneira com que as séries temporais de treinamento são geradas. Assim, em teoria, seria possível utilizar séries temporais geradas a partir de estados iniciais governados por diferentes distribuições de probabilidade (a forma da distribuição continuaria sendo Gaussiana, mas as médias e/ou covariâncias seriam diferentes), obtendo então sintonias que seriam capazes de lidar com uma ampla faixa de condições de operação do filtro.

Dependendo do sistema dinâmico, porém, isso pode não ser necessário. A sintonia obtida com dados de treinamento gerados a partir de uma única distribuição do estado inicial poderia ser capaz de fazer o filtro funcionar bem em outras regiões do espaço de estados do sistema dinâmico. Para verificar essa hipótese, os experimentos apresentados nesse capítulo submetem os filtros sintonizados nos experimentos descritos nos Capítulos 9 e 10 a séries de teste geradas a partir de distribuições de estado inicial com médias diferentes daquelas em que foram gerados os dados utilizados na sintonia dos filtros.

11.1 Descrição dos Experimentos

Para cada um dos problemas de filtragem descritos no Capítulo 8, são realizados N_e experimentos independentes, correspondendo a N_e distribuições distintas do estado inicial.

Os critérios MSEx e NCI são calculados para cada um dos experimentos, gerando assim N_e valores do MSEx e N_e valores do NCI para cada experimento. O desempenho de um filtro em um dado problema de filtragem é dado pelo valor médio e pelo desvio padrão dos MSExs e NCIs calculados sobre os N_e experimentos.

11.2 Kitagawa

11.2.1 Dados de Teste

Para cada distribuição do estado inicial, foram geradas 100 séries temporais por simulação de Monte Carlo. As distribuições usadas foram $\mathbf{x}_0 = \mathcal{N}(\mu, \sigma^2)$, com $\mu \in \{-4, -2, 0, 2, 4\}$ e $\sigma^2 = 0.25$.

11.2.2 Resultados Numéricos

Os desempenhos dos filtros são mostrados nas Tabelas 11.1, 11.2, 11.3 e 11.4.

Tabela 11.1: *Kitagawa - Valores Computados em 5 Experimentos - Ruídos Conhecidos*

	UkfD	CKF	UkfO	Ukf-ML	EKF	UkfA	PfUkf
Média MSEx	0.404	13.004	0.091	0.355	27.883	0.083	0.082
Desvio Padrão MSEx	0.376	7.854	0.053	0.577	34.096	0.038	0.037
Média NCI	1.195	54.947	1.347	11.243	52.179	1.419	1.286
Desvio Padrão NCI	3.562	5.976	3.997	3.350	3.615	4.052	3.853

Tabela 11.2: *Kitagawa - Classificação em 5 Experimentos - Ruídos Conhecidos*

	1	2	3	4	5	6	7
Média MSEx	PfUkf	UkfA	UkfO	Ukf-ML	UkfD	CKF	EKF
Média NCI	UkfD	PfUkf	UkfO	UkfA	Ukf-ML	EKF	CKF

Tabela 11.3: *Kitagawa - Valores Computados em 5 Experimentos - Ruídos Identificados*

	UkfD	CKF	UkfO_QR	Ukf-ML	EKF	UkfA_QR	PfUkf_QR
Média MSEx	0.404	13.004	0.087	0.355	27.883	0.084	0.086
Desvio Padrão MSEx	0.376	7.854	0.045	0.577	34.096	0.038	0.044
Média NCI	1.195	54.947	1.630	11.243	52.179	0.799	1.525
Desvio Padrão NCI	3.562	5.976	3.974	3.350	3.615	3.949	3.537

Tabela 11.4: *Kitagawa - Classificação em 5 Experimentos - Ruídos Identificados*

	1	2	3	4	5	6	7
Média MSEx	UkfA_QR	PfUkf_QR	UkfO_QR	Ukf-ML	UkfD	CKF	EKF
Média NCI	UkfA_QR	UkfD	PfUkf_QR	UkfO_QR	Ukf-ML	EKF	CKF

A Figura 11.1 exibe a distribuição dos MSEx calculados em cada um dos 5 experimentos realizados para Kitagawa no caso de ruídos conhecidos, enquanto a Figura 11.2 exibe a distribuição para o caso de ruídos identificados.

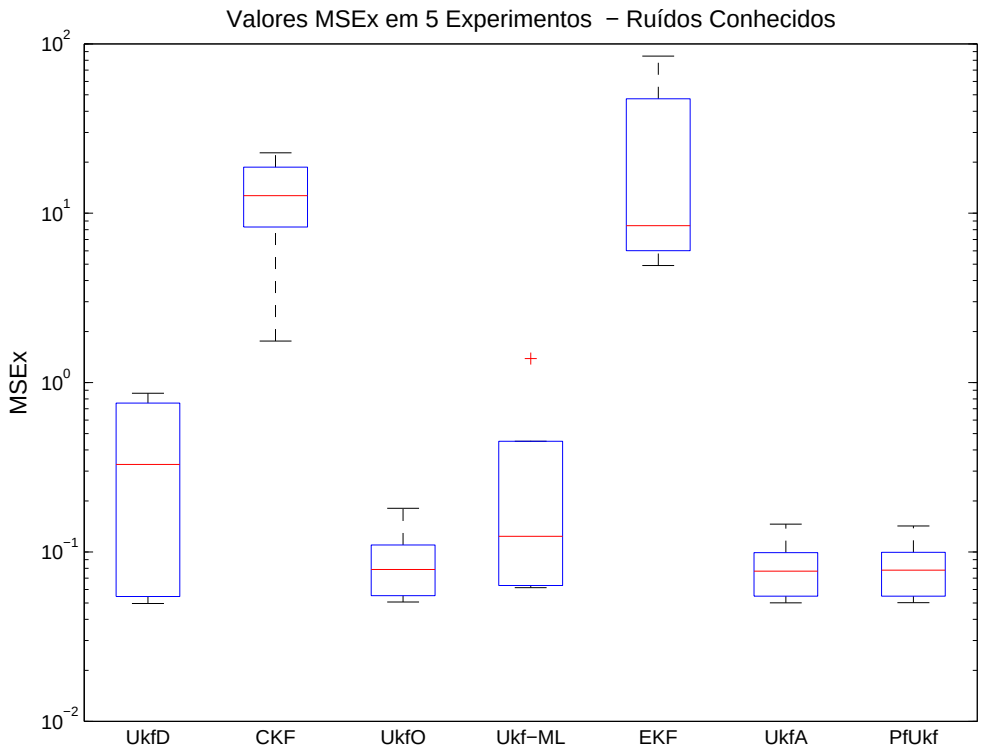

Figura 11.1: *Kitagawa - Múltiplos Experimentos - MSEx com Ruídos Conhecidos*

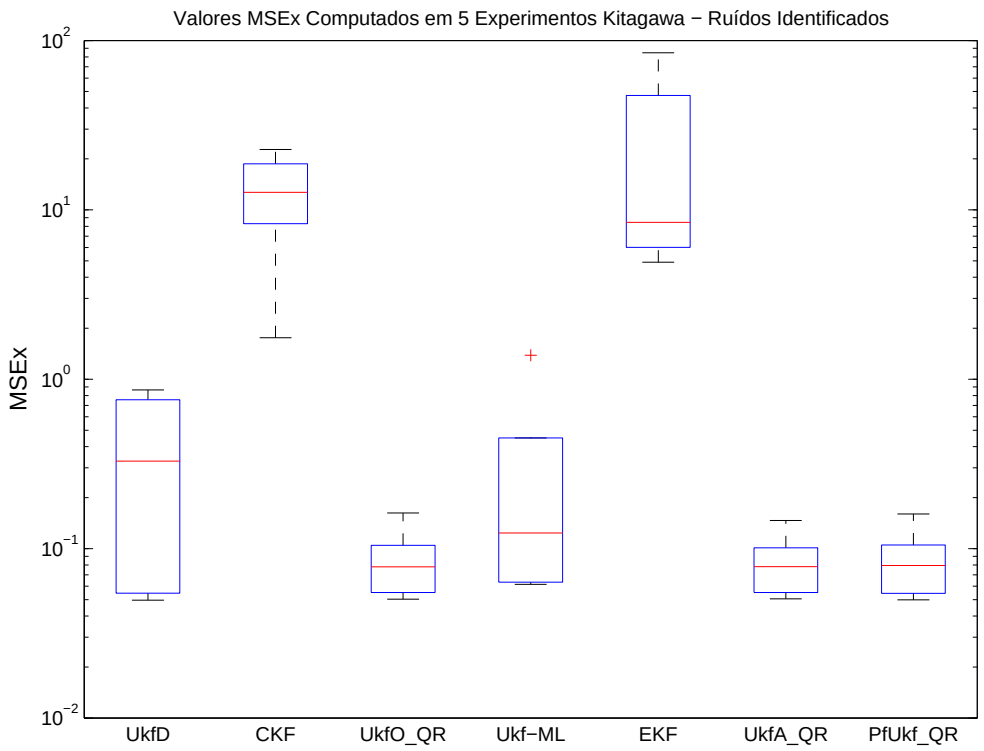

Figura 11.2: *Kitagawa - Múltiplos Experimentos - MSEx com Ruídos Identificados*

11.2.3 Análise

Nos experimentos com ruídos conhecidos, os filtros PfUkf e UkfA apresentaram resultados quase MSEx idênticos. Ambos foram melhores que o UkfO. Os três filtros apresentaram credibilidades semelhantes, próximas à credibilidade do UkfD. A situação praticamente se repetiu nos experimentos com ruídos identificados.

Em ambas as situações, o Ukf-ML foi pouco melhor que o UkfD, enquanto o CKF e o EKF apresentaram desempenhos insatisfatórios.

11.3 Sinusoid

11.3.1 Dados de Teste

Para cada distribuição do estado inicial, foram geradas 100 séries temporais por simulação de Monte Carlo. As distribuições usadas foram $\mathbf{x}_0 = \mathcal{N}(\mu, \sigma^2)$, com $\mu \in \{-4, -2, 0, 2, 4\}$, and $\sigma^2 = 0.25$.

11.3.2 Resultados Numéricos

Tabela 11.5: *Sinusoid - Valores Computados em 5 Experimentos - Ruídos Conhecidos*

	UkfD	CKF	UkfO	Ukf-ML	EKF	UkfA	PfUkf
Média MSEx	0.966	1.460	0.760	1.410	1.546	0.766	0.788
Desvio Padrão MSEx	0.006	0.095	0.007	0.103	0.041	0.027	0.005
Média NCI	-0.761	7.184	0.349	2.151	3.050	0.701	0.644
Desvio Padrão NCI	0.637	1.497	0.707	1.000	1.058	0.795	0.718

Tabela 11.6: *Sinusoid - Classificação em 5 Experimentos - Ruídos Conhecidos*

	1	2	3	4	5	6	7
Média MSEx	UkfO	UkfA	PfUkf	UkfD	Ukf-ML	CKF	EKF
Média NCI	UkfO	PfUkf	UkfA	UkfD	Ukf-ML	EKF	CKF

Tabela 11.7: *Sinusoid - Valores Computados em 5 Experimentos - Ruídos Identificados*

	UkfD	CKF	UkfO_QR	Ukf-ML	EKF	UkfA_QR	PfUkf_QR
Média MSEx	0.966	1.460	0.975	1.410	1.546	0.802	0.831
Desvio Padrão MSEx	0.006	0.095	0.006	0.103	0.041	0.046	0.006
Média NCI	-0.761	7.184	-0.473	2.151	3.050	1.348	-0.685
Desvio Padrão NCI	0.637	1.497	0.640	1.000	1.058	0.677	0.590

Tabela 11.8: *Sinusoid - Classificação em 5 Experimentos - Ruídos Identificados*

	1	2	3	4	5	6	7
Média MSEx	UkfA_QR	PfUkf_QR	UkfD	UkfO_QR	Ukf-ML	CKF	EKF
Média NCI	UkfO_QR	PfUkf_QR	UkfD	UkfA_QR	Ukf-ML	EKF	CKF

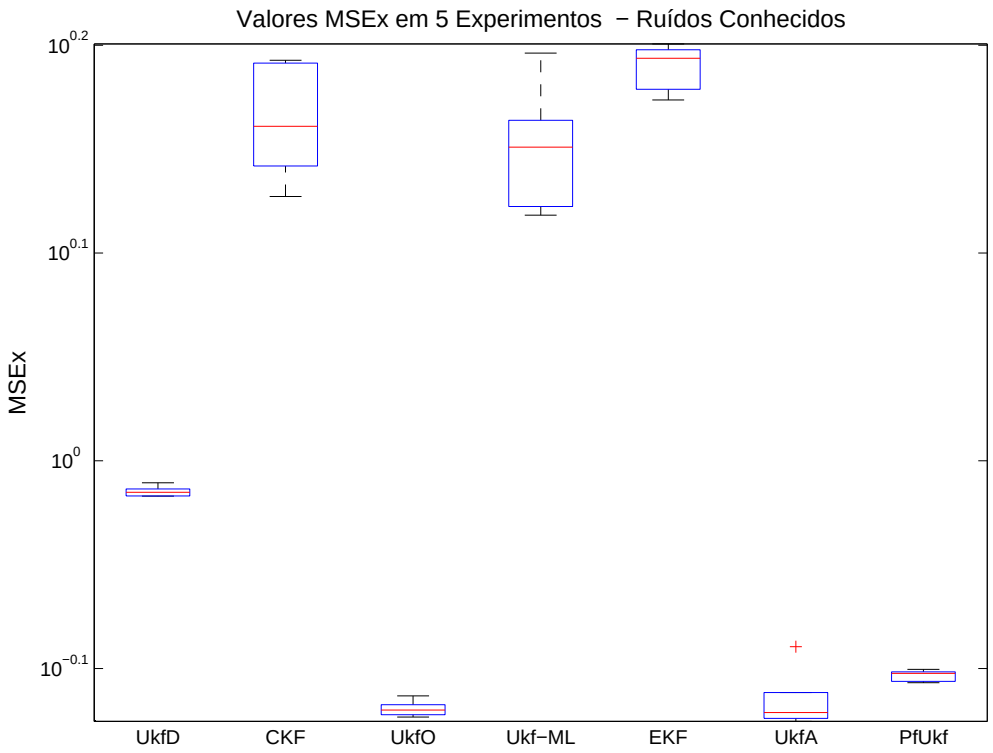

Figura 11.3: *Sinusoid - Múltiplos Experimentos - MSEx com Ruídos Conhecidos*

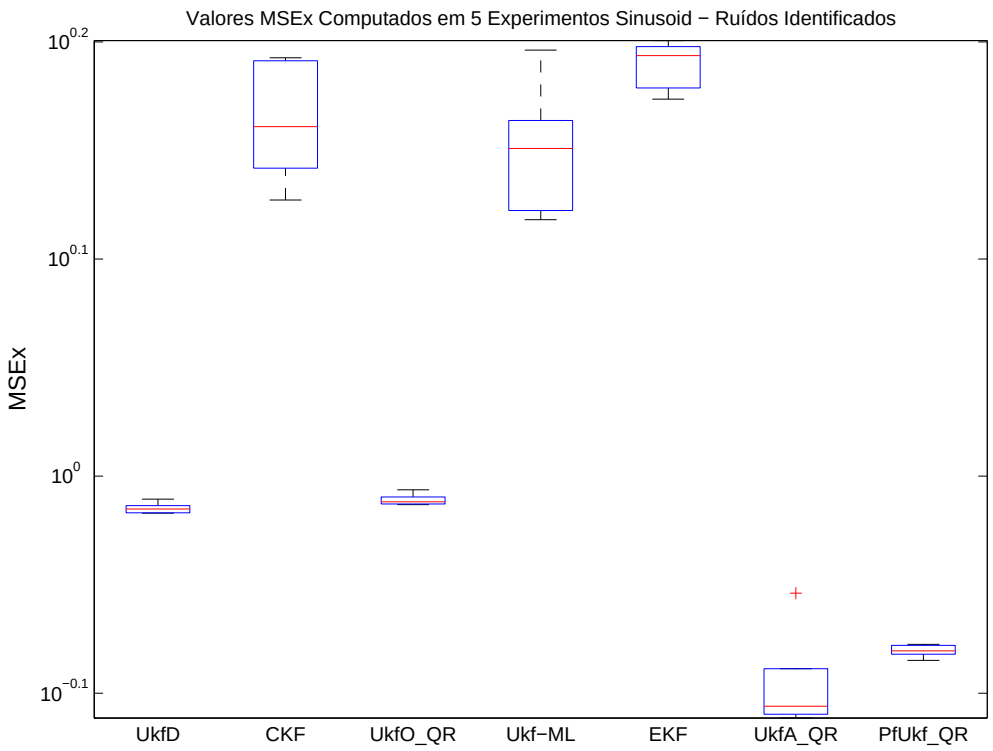

Figura 11.4: *Sinusoid - Múltiplos Experimentos - MSEx com Ruídos Identificados*

11.3.3 Análise

Nos experimentos com ruídos conhecidos, em termos de MSEx, o melhor filtro foi UkfO, seguido de perto pelo UkfA e pelo PfUkf. Em termos de credibilidade, UkfO foi o melhor filtro, seguido pelo PfUkf e pelo UkfA.

Nos experimentos com ruídos identificados, em termos de MSEx, o UkfA-QR e o PfUkf-QR foram melhores que o UkfO-QR. Em termos de credibilidade, UkfO-QR foi o melhor filtro, seguido do PfUkf-QR.

11.4 Bearings Only

11.4.1 Dados de Teste

Para cada distribuição do estado inicial, foram geradas 100 séries temporais por simulação de Monte Carlo. As distribuições usadas foram $\mathbf{x}_0 = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, com

$$\boldsymbol{\mu} \in \left\{ \begin{pmatrix} 16 \\ 1 \end{pmatrix}, \begin{pmatrix} 18 \\ 3 \end{pmatrix}, \begin{pmatrix} 20 \\ 5 \end{pmatrix}, \begin{pmatrix} 22 \\ 7 \end{pmatrix}, \begin{pmatrix} 24 \\ 9 \end{pmatrix} \right\} \tag{11.1}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} . \tag{11.2}$$

11.4.2 Resultados Numéricos

Tabela 11.9: *Bearings - Valores Computados em 5 Experimentos - Ruídos Conhecidos*

	UkfD	CKF	UkfO	Ukf-ML	EKF	UkfA	PfUkf
Média MSEx	11.918	19.208	3.389	1.774	53.952	2.630	2.702
Desvio Padrão MSEx	8.131	9.928	1.458	0.779	23.392	0.850	0.697
Média NCI	8.518	9.405	1.757	0.549	14.453	1.051	0.765
Desvio Padrão NCI	5.796	6.372	1.613	0.605	9.166	0.801	0.709

Tabela 11.10: *Bearings - Classificação em 5 Experimentos - Ruídos Conhecidos*

	1	2	3	4	5	6	7
Média MSEx	Ukf-ML	UkfA	PfUkf	UkfO	UkfD	CKF	EKF
Média NCI	Ukf-ML	PfUkf	UkfA	UkfO	UkfD	CKF	EKF

Tabela 11.11: *Bearings Only - Valores Computados em 5 Experimentos - Ruídos Identificados*

	UkfD	CKF	UkfO_QR	Ukf-ML	EKF	UkfA_QR	PfUkf_QR
Média MSEx	11.918	19.208	2.784	1.774	53.952	2.241	2.774
Desvio Padrão MSEx	8.131	9.928	1.194	0.779	23.392	0.818	1.131
Média NCI	8.518	9.405	0.346	0.549	14.453	-0.816	0.747
Desvio Padrão NCI	5.796	6.372	1.357	0.605	9.166	0.385	1.509

Tabela 11.12: *Bearings Only - Classificação em 5 Experimentos - Ruídos Identificados*

	1	2	3	4	5	6	7
Média MSEx	Ukf-ML	UkfA_QR	PfUkf_QR	UkfO_QR	UkfD	CKF	EKF
Média NCI	UkfO_QR	Ukf-ML	PfUkf_QR	UkfA_QR	UkfD	CKF	EKF

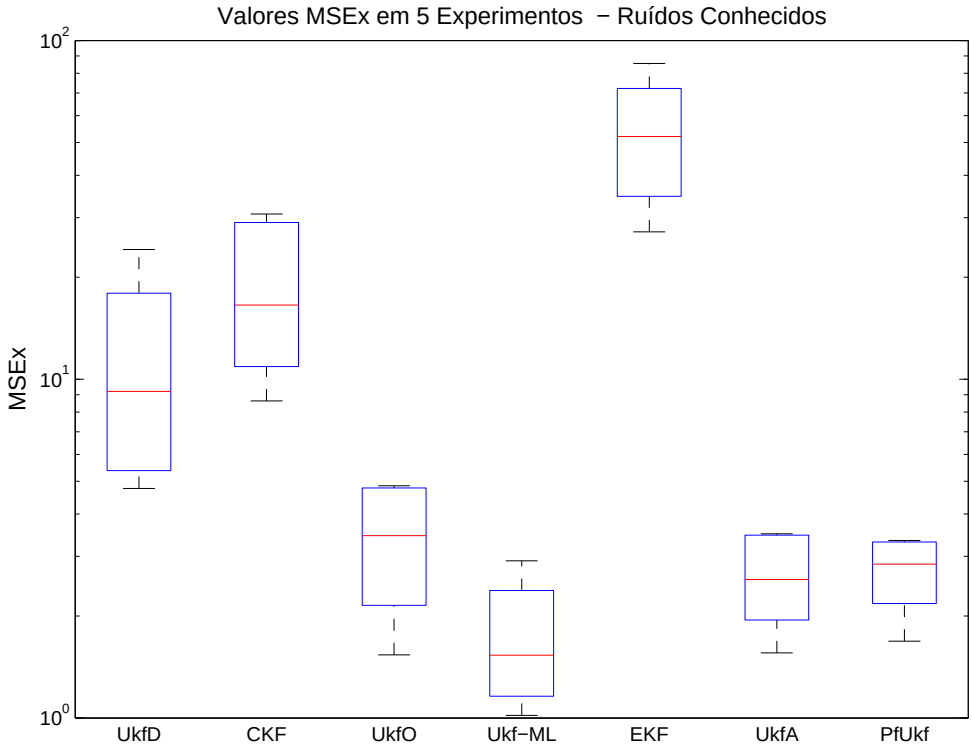

Figura 11.5: *Bearings Only - Múltiplos Experimentos - MSEx com Ruídos Conhecidos*

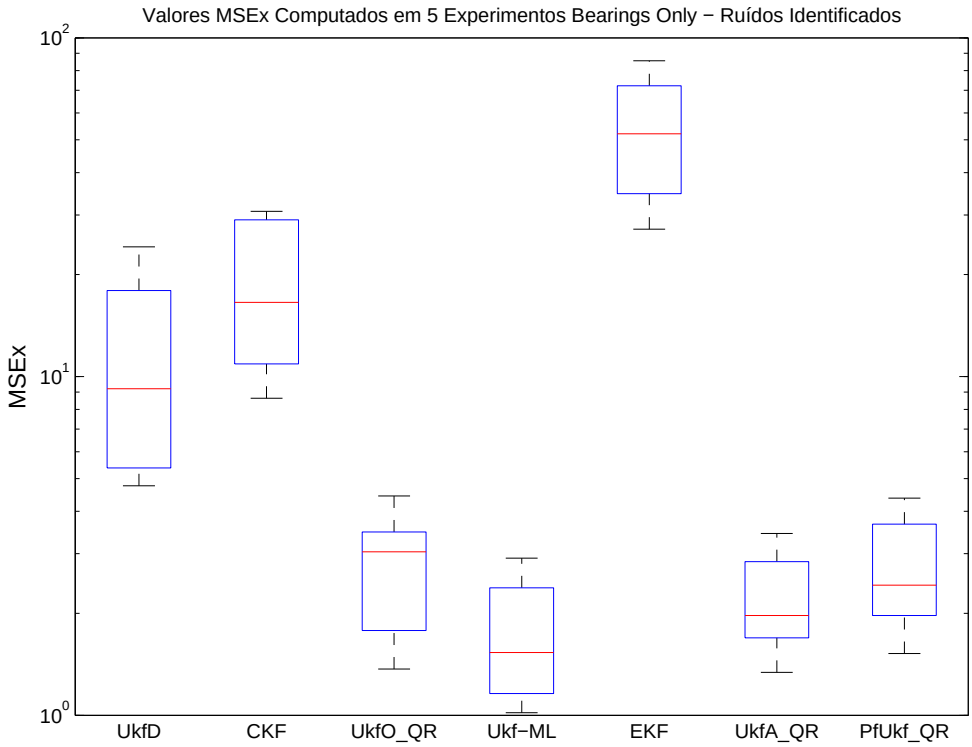

Figura 11.6: *Bearings Only - Múltiplos Experimentos - MSEx com Ruídos Identificados*

11.4.3 Análise

Nos experimentos com ruídos conhecidos, o Ukf-ML foi o melhor filtro, seguido pelo UkfA, pelo PfUkf e pelo UkfO. Os demais filtros tiveram desempenhos insatisfatórios. Novamente os melhores filtros em termos de MSEx foram os melhores em termos de NCI, embora não tenha havido correspondência exata entre as classificações feitas pelos dois critérios. Nos experimentos com ruídos identificados, a ordem em termos de MSEx se manteve.

Capítulo 12

O Problema *Blind Tricyclist*

Descrevemos agora a aplicação dos algoritmos propostos no Capítulo 7 a um problema de filtragem não-linear recentemente proposto, o problema do ciclista cego (blind tricyclist) [56]. O problema apresenta não linearidades significativas, tanto no modelo de processo quando no modelo de medidas. Tais não linearidades, juntamente com o fato de o vetor de estados ser composto por $3 + 2M$ variáveis[1], implicam significativas dificuldades para os algoritmos de filtragem. Nosso interesse no problema resulta principalmente do fato de [56] afirmar que, quando a incerteza a respeito do estado inicial é alta, o UKF pode apresentar desempenho significativamente pior que o EKF.

Um aspecto relevante do uso do UKF, que não é muito trabalhado no estudo apresentado em [56], é a escolha dos valores para os parâmetros da UT. Assim, neste capítulo, apresentamos o processo de sintonização do UKF aplicado ao problema do ciclista cego. Para permitir comparação com os resultados apresentados em [56], utilizamos as funções MATLAB fornecidos pelos autores em [60], para gerar todos os dados de treinamento e de teste.

12.1 O Problema

Neste problema, uma pessoa cega, que pilota um triciclo, deseja navegar por um parque de diversões. A única informação da qual dispõe para navegar advém de pessoas sentadas em carrosséis. Essas pessoas gritam de forma intermitente para o ciclista, provendo, com isso, informações de direção relativa. A tarefa de navegação é complicada pelo fato de o ciclista ter de estimar o ângulo de rotação inicial e a taxa de rotação constante de cada carrossel. A necessidade de estimar esses parâmetros confere ao problema a característica de um problema de localização e mapeamento simultâneos (SLAM). Como veremos na breve descrição que faremos, as equações de processo e de medida do modelo apresentam não-linearidades significativas. A descrição detalhada do problema é extensa e complexa, podendo ser encontrada em [56], de onde foi extraída a Figura 12.1.

[1] Nos nossos experimentos, $M = 2$, indicando que há duas pessoas, cada uma sentada em um carrossel diferente.

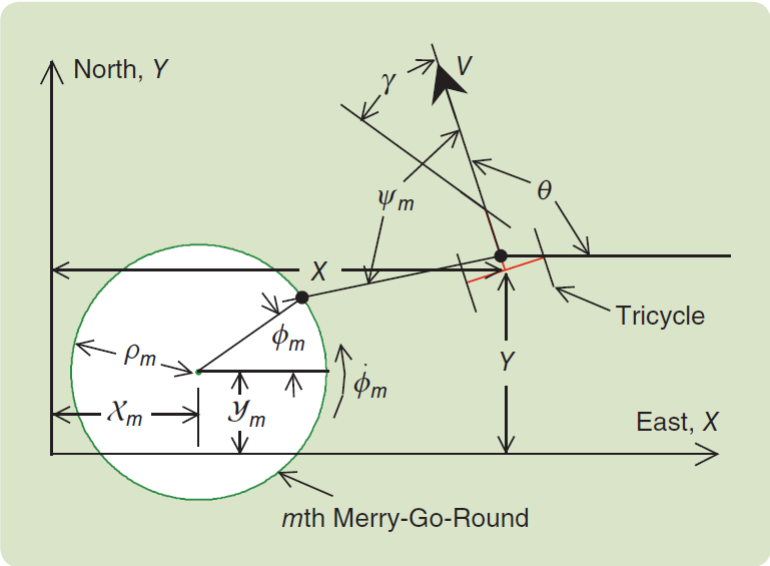

Figura 12.1: *Geometria do problema Blind Tricyclist*

O vetor de estado é dado por

$$\mathbf{x} = [X, Y, \theta, \phi_1, \ldots, \phi_M, \dot{\phi}_1, \ldots, \dot{\phi}_M]^T \tag{12.1}$$

onde X, Y são as localizações norte e leste e θ é o ângulo de direção do ciclista, ϕ_m é o ângulo de rotação do m-ésimo carrossel, e M é o número de carrosséis diferentes.

O vetor de controles bidimensional é dado por

$$\mathbf{u} = [V, \gamma]^T \tag{12.2}$$

onde V e γ são respectivamente a velocidade e o ângulo comandados.

Os modelos dinâmicos para os elementos do vetor de estado são

$$\begin{aligned} X_{k+1} = X_k + (V_k + w_{1k})\Delta t[sin\theta_k cinc\{\frac{\Delta t(V_k + w_{1k})tan(\gamma_k + w_{2k})}{b_w}\} \\ + cos\theta_k sinc\{\frac{\Delta t(V_k + w_{1k})tan(\gamma_k + w_{2k})}{b_w}\}] + \Delta t w_{3k} \end{aligned} \tag{12.3a}$$

$$\begin{aligned} Y_{k+1} = Y_k + (V_k + w_{1k})\Delta t[-cos\theta_k cinc\{\frac{\Delta t(V_k + w_{1k})tan(\gamma_k + w_{2k})}{b_w}\} \\ + sin\theta_k sinc\{\frac{\Delta t(V_k + w_{1k})tan(\gamma_k + w_{2k})}{b_w}\}] + \Delta t w_{4k} \end{aligned} \tag{12.3b}$$

$$\theta_{k+1} = \theta_k + \frac{\Delta t(V_k + w_{1k})tan(\gamma_k + w_{2k})}{b_w} + \Delta t w_{5k} \tag{12.3c}$$

$$\begin{aligned} \phi_{mk+1} &= \phi_{mk} + \dot{\phi}_{mk}\Delta t \quad \text{for } m \\ &= 1, \ldots, M \end{aligned} \tag{12.3d}$$

$$\begin{aligned} \dot{\phi}_{mk+1} &= \dot{\phi}_{mk} \quad \text{for } m \\ &= 1, \ldots, M \ , \end{aligned} \tag{12.3e}$$

onde

- $\mathbf{x}_k$ é o vetor de estado do sistema no instante de amostragem $t_k = k\Delta t$

- Δt é a duração do intervalo de amostragem
- ρ_m é o raio do m-ésimo carrossel
- $cinc(\alpha) = \frac{cos(\alpha)-1}{\alpha}$ e $sinc(\alpha) = \frac{sin(\alpha)}{\alpha}$
- V_k e γ_k são respectivamente a velocidade e o ângulo comandados
- $w_{1k}, \ldots, w_{5k}$ são componentes do vetor de ruído de processo $\mathbf{w}_k$
- b_w é a distância entre as rodas de trás do triciclo.

O ruído de processo é modelado como ruído Gaussiano branco de tempo discreto, com as seguintes estatísticas

$$E\{\mathbf{w}_k\} = \mathbf{0} \tag{12.4}$$

$$E\{\mathbf{w_k w_j^T}\} = \begin{Bmatrix} \mathbf{0} & \text{if } j \neq k \\ \mathbf{Q}_k & \text{if } j = k \end{Bmatrix} \tag{12.5}$$

sendo a matriz de covariância $\mathbf{Q}_k$ conhecida.

Tomando como referência a Figura 12.1, a orientação relativa do m-ésimo amigo no m-ésimo carrossel é dada por

$$\psi_{mk} = atan2\{(\mathcal{Y}_m + \rho_m \sin\phi_{mk} - Y_k - b_r \sin\theta_k), (\mathcal{X}_m + \rho_m \cos\phi_{mk} - X_k - b_r \cos\theta_k)\} - \theta_k + v_{mk} \ , \tag{12.6}$$

onde $\psi_{mk} := \psi_m(t_k)$, b_r é a distância do ponto abaixo da cabeça do ciclista ao ponto médio entre as rodas de trás, e v_{mk} é o ruído de medida da orientação relativa, que é considerado ruído branco Gaussiano de tempo discreto com as seguintes estatísticas

$$E\{v_{mk}\} = 0, \quad E\{v_{mk} v_{mj}\} = \begin{Bmatrix} 0 & \text{para } j \neq k \\ \sigma_m^2 & \text{para } j = k \end{Bmatrix} \tag{12.7}$$

$$E\{v_{mk} v_{lj}\} = 0 \quad \text{para todo } j \text{ e } k, \text{ para } l \neq m \tag{12.8}$$

Assume-se assim que os erros de medida da orientação relativa em relação a carrosséis diferentes são não correlacionados.

A forma geral das equações do modelo que geram os dados para o problema é dada por

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \tag{12.9a}$$

$$\mathbf{y}_k = h_k(\mathbf{x}_k) + \mathbf{v}_k \ . \tag{12.9b}$$

O ruído de medida também é modelado como ruído branco Gaussiano de tempo discreto, com as seguintes estatísticas

$$E\{\mathbf{v}_k\} = \mathbf{0}, \quad E\{\mathbf{v_k v_j^T}\} = \begin{Bmatrix} \mathbf{0} & \text{if } j \neq k \\ \mathbf{R}_k & \text{if } j = k \end{Bmatrix} \ , \tag{12.10}$$

onde a matriz de covariância $\mathbf{R}_k$ é uma matriz diagonal conhecida.

A implementação do UKF usada nesse trabalho assume modelo dinâmico em que os ruídos são aditivos, como pode ser visto na equação (3.5). Assim, foi necessário adaptar a matriz de covariância do ruído de processo $\mathbf{Q}_k$ multiplicando-a pela matriz matrix Γ_k, o Jacobiano da equação de processo do modelo em relação aos elementos do ruído de processo. Essa estratégia é a mesma utilizada no exemplo de filtragem com o EKF que é provido em [60]. A matriz de covariância resultante é dada por $\Gamma_k \mathbf{Q}_k (\Gamma_k^T)$. O leitor interessado pode encontrar mais detalhes na documentação provida em [60].

12.2 Experimento para o Ciclista Cego

Para os Algoritmos 3 e 5, os parâmetros livres foram dados por:

- $\mathbf{\Theta} \leftarrow 27$ amostras geradas como grade de pontos no cubo definido por Range
- $N_s \leftarrow 10$
- Range

  $0.01 \leq \alpha \leq 4$

  $0 \leq \beta \leq 4$

  $0 \leq \kappa \leq 5$
- budget $\leftarrow 100$

12.2.1 Dados de Treinamento

O processo de sintonia dos filtros UkfA, UkfO e PfUkf usou uma série temporal de treinamento, gerada por simulação do modelo do sistema usando os códigos Matlab providos em [60]

12.2.2 Dados de Teste

Para o teste, 100 séries temporais foram geradas por simulação de Monte Carlo do modelo dinâmico, novamente usando os códigos Matlab fornecidos em [60].

12.2.3 Sintonia

A sintonia do UkfA usou o critério UCB (7.3). A evolução dos processos de sintonia dos filtros PfUkf, UkfA e UkfO é mostrada na Figura 12.2. As sintonias resultantes são apresentadas na Tabela 12.1

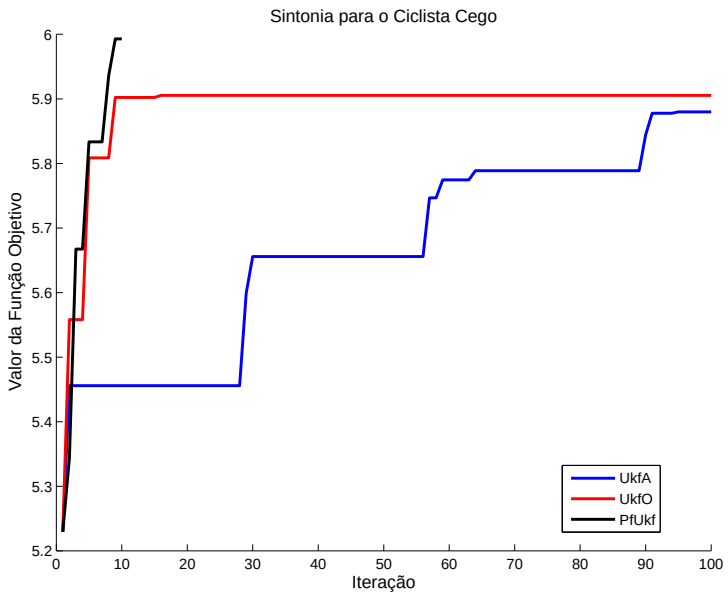

Figura 12.2: *Blind Tricyclist - Sintonia dos Filtros*

Tabela 12.1: *Blind Tricyclist - Parâmetros da UT*

	α	β	κ
UkfO	0.34872	3.13828	1.40269
UkfA	0.31821	3.36424	1.62577
PfUkf	0.32277	1.85289	1.59410

12.2.4 Resultados Numéricos

Os erros de estimação de estado são mostrados na Tabela 12.2. A primeira linha da tabela apresenta o RMSE (medido em metros) nas estimativas da posição final do ciclista, enquanto a segunda linha apresenta o módulo da variação correspondente ao intervalo de confiança de 95% para esse erro. A terceira linha apresenta o tempo médio de computação de cada filtro (em segundos). A distribuição dos valores do erro médio quadrático das estimativas da posição final do ciclista é mostrada na Figura 12.3.

Tabela 12.2: *Blind Tricyclist - Erros nas Estimativas da Posição Final*

	UkfA	UkfO	Ckf	Ekf	Ukf-ML	PfUkf
RMSE Posição Final (m)	2.389	3.766	81.174	6.850	72.936	2.467
$\Delta_{.95X}$	1.918	3.259	64.079	3.731	49.672	1.963
Tempo Médio (s)	0.594	0.585	0.587	0.065	10.502	0.604

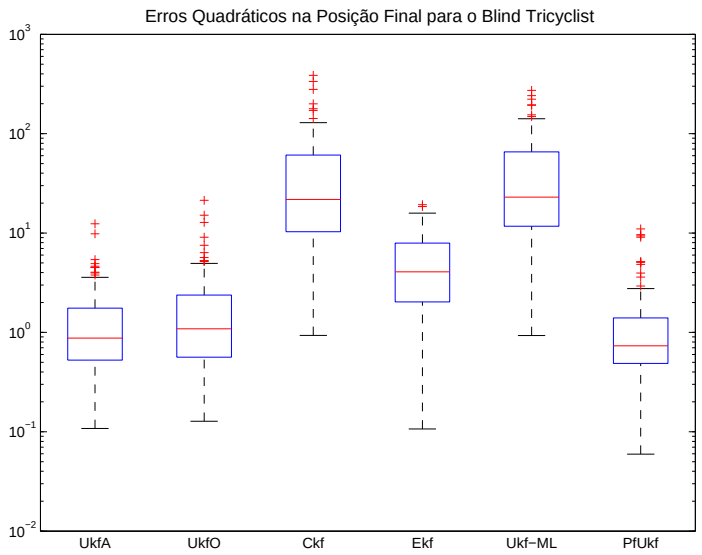

Figura 12.3: *Blind Tricyclist - Erros nas Estimativas da Posição Final*

A evolução temporal dos RMSE nas estimativas de posição do ciclista computados nos dados de teste são mostrados na Figura 12.4.

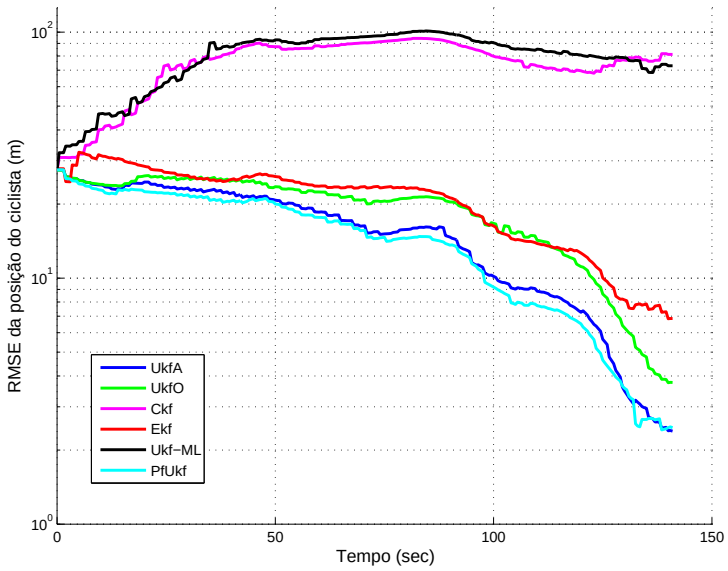

Figura 12.4: *Blind Tricyclist - Evolução Temporal dos Erros nas Estimativas de Posição*

A evolução temporal da credibilidade dos filtros calculada nos dados de teste é apresentada na Figura 12.5.

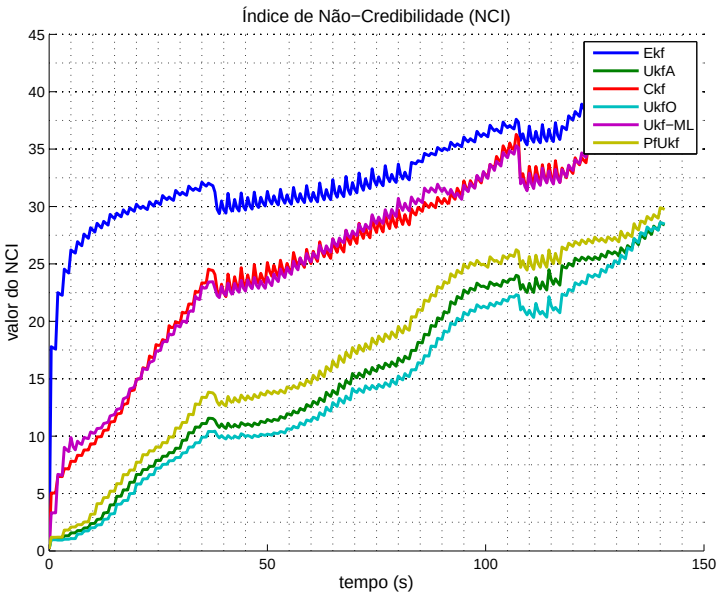

Figura 12.5: *Blind Tricyclist - Índices de Não-Credibilidade*

12.2.5 Análise

Em termos da qualidade das estimativas de estado, o UkfA e o PfUkf foram claramente os melhores filtros. Em terceiro lugar, um pouco mais distante, veio o UkfO. O EKF, que em [56] superou por larga margem o UKF, foi aqui apenas o quarto melhor filtro, enquanto o Ukf-ML e o CKF apresentaram novamente desempenhos muito inferiores aos dos melhores filtros. A Figura 12.3 mostra que o CKF e o Ukf-ML produziram valores RMSE consistentemente altos.

A melhora do desempenho do UKF em relação ao desempenho reportado em [56] se deveu essencialmente à sintonia dos parâmetros da UT. Vale lembrar que a hipótese de que seria possível melhorar o desempenho do UKF por meio de ajustes nos parâmetros da UT foi levantada pelo próprio autor de [56]. Os resultados aqui obtidos corroboram essa hipótese.

Em termos de confiabilidade, os três filtros que utilizaram sintonia *offline* apresentaram os melhores valores, enquanto EKF, CKF e Ukf-ML apresentaram as piores credibilidades.

Os resultados obtidos permitem concluir que a sintonia é fundamental para o bom funcionamento do UKF.

Capítulo 13

Conclusões e Trabalhos Futuros

Tratamos neste trabalho da sintonia do UKF. O problema é relevante, dadas a significativa importância do filtro e a dificuldade da tarefa de efetuar manualmente a sintonia. Aspecto importante dos algoritmos de sintonia propostos é o fato de poderem sintonizar o filtro com base somente em medidas ruidosas do estado do sistema dinâmico. Esses algoritmos não demandam a existência de medidas de alta precisão. Também não demandam que sejam conhecidos os estados reais que deram origem às medidas usadas no processo de sintonia. Desse modo, os algoritmos propostos são aplicáveis a uma ampla gama de situações.

Outro aspecto relevante dos algoritmos de sintonia propostos é o fato de efetuaram o processo de sintonia *offline*, o que não traz nenhum custo computacional extra para o UKF. Isso permite que o UKF sintonizado mantenha um dos principais atrativos do filtro, que é a baixa complexidade computacional.

De modo geral, os resultados numéricos indicam que os algoritmos de sintonia propostos foram capazes de lidar com problemas de filtragem caracterizados por não linearidades significativas, que quase sempre fizeram com que os filtros não sintonizados produzissem estimativas de estado de baixa qualidade.

Em termos da qualidade das estimativas de estado produzidas, o UKF resultante da sintonia com base em partículas (Algoritmo 3) foi o melhor filtro em 3 dos 13 experimentos numéricos realizados, estando sempre entre os 3 melhores filtros. Esses números são surpreendentes, dada a simplicidade do Algoritmo 3. O UKF resultante da sintonia baseada em modelos (Algoritmo 5) foi o melhor em 5 dos 13 experimentos, também estando sempre entre os 3 melhores filtros. Nenhum dos demais filtros apresentou resultados tão consistentes quanto os filtros sintonizados pelos algoritmos propostos nesse trabalho.

Quanto ao impacto da sintonia na credibilidade do filtro, os resultados numéricos indicaram que, em geral, os menores índices de não-credibilidade estiveram associados aos filtros que obtiveram os melhores resultados em termos da estimação de estados. Assim, é possível dizer que o processo de sintonia *offline* contribuiu para aumentar a credibilidade do UKF.

Os experimentos numéricos realizados nesse trabalho também indicaram que o processo de sintonia *offline* pode gerar sintonias capazes de lidar com estados iniciais diferentes daqueles usados no processo de sintonia. Mesmo assim, é razoável esperar que haja sistemas dinâmicos para os quais a sintonia teria de ser refeita quando a média e/ou covariância da distribuição de probabilidade do estado inicial fosse muito diferente daquela utilizada para gerar as séries de treinamento. Haveria então oportunidade de pesquisa no estudo de mecanismos voltados para a ampliação da faixa de utilização das sintonias obtidas *offline*.

Por exemplo, é razoável supor que a faixa de utilização da sintonia obtida *offline* esteja ligada à representatividade das séries temporais usadas para a sintonia dos filtros. Quanto mais as séries temporais usadas no processo de sintonia forem representativas das situações

passíveis de serem encontradas pelo filtro, maiores as possibilidades de o filtro sintonizado produzir estimativas de estado satisfatórias quando estiver sendo aplicado ao problema real. Dentre os elementos que afetam a representatividade de uma série de treinamento estão, por exemplo, a distribuição de probabilidade do estado inicial e a duração da série.

Também seria relevante o desenvolvimento de funções objetivo capazes de ponderar de forma diferente as séries temporais que recebe, em função da representatividade de cada série. Isso poderia trazer grandes ganhos para o processo de sintonia, pois evitaria que uma série temporal de baixa representatividade tivesse o mesmo peso que uma série de alta representatividade.

Outro desdobramento possível seria a aplicação das técnicas de sintonia propostas a situações nas quais não se dispõe de modelo preciso do sistema dinâmico. Nesses casos, a sintonia simultânea dos parâmetros da UT e das matrizes de covariância dos ruídos de processo e de medidas poderia ser usada para compensar falhas de modelagem.

Diante do que foi exposto, podemos afirmar que os algoritmos propostos puderam melhorar a qualidade das estimativas de estado produzidas pelo UKF, por meio da sintonia dos parâmetros do filtro. Fizeram isso sem impor custo computacional extra ao filtro sintonizado, permitindo assim que o UKF preservasse sua baixa complexidade computacional, um dos grandes atrativos desse filtro. Desse modo, embora haja espaço para novas pesquisas no problema de sintonia do UKF, podemos concluir que esse trabalho alcançou as metas inicialmente estabelecidas, tendo prestado contribuição efetiva para a área de filtragem não linear.

Referências Bibliográficas

[1] J. Dunik, M. Simandl, and O. Straka, "Unscented Kalman filter: Aspects and adaptive setting of scaling parameter." *IEEE Trans. Automat. Contr.*, vol. 57, no. 9, pp. 2411–2416, 2012. ix, 16, 17, 19, 21, 32, 35

[2] O. Straka, J. Dunik, and M. Simandl, "Scaling parameter in unscented transform: Analysis and specification," in *American Control Conference (ACC), 2012*, june 2012, pp. 5550 –5555. ix, 19, 32

[3] ——, "Unscented Kalman filter with advanced adaptation of scaling parameter," *Automatica*, vol. 50, no. 10, pp. 2657 – 2664, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000510981400346X ix, 16, 17, 19, 32

[4] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, March 2000. ix, 11, 17, 31

[5] R. Turner, "Code for mlsp 2010," http://mlg.eng.cam.ac.uk/rdturner/code.html, 2011, accessed November 4, 2014. ix, 30, 32, 36, 37

[6] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *American Control Conference, Proceedings of the 1995*, vol. 3, 1995, pp. 1628–1632 vol.3. 1

[7] R. Van Der Merwe, "Sigma-point Kalman filters for probabilistic inference in dynamic state-space models," Ph.D. dissertation, 2004. 1

[8] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," in *Proceedings of the IEEE*, vol. 3, no. 92, 2004, pp. 401–422. 1, 31

[9] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, Apr 1993. 1, 22

[10] S. J. Julier and I. Industries, "The scaled unscented transformation," in *in Proc. IEEE Amer. Control Conf*, 2002, pp. 4555–4559. 1

[11] D. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization*, vol. 21, no. 4, pp. 345–383, 2001. 1, 27

[12] Y. Ho and R. Lee, "A Bayesian approach to problems in stochastic estimation and control," *Automatic Control, IEEE Transactions on*, vol. 9, no. 4, pp. 333–339, Oct 1964. 2

[13] A. H. Jazwinski, *Stochastic Processes and Filtering Theory.* Academic Press, 1970. 2, 6, 31

[14] S. Sarkka, *Bayesian Filtering and Smoothing.* Cambridge University Press, 2013. 2, 3, 7, 10, 12

[15] J. V. Candy, *Bayesian Signal Processing: Classical, Modern and Particle Filtering Methods.* New York, NY, USA: Wiley-Interscience, 2009. 2

[16] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *Automatic Control, IEEE Transactions on*, vol. 17, no. 4, pp. 439–448, 1972. 2

[17] H. Sorenson, "On the development of practical nonlinear filters," *Information Sciences*, vol. 7, no. 0, pp. 253 – 270, 1974. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0020025574900176 5

[18] S. Sarkka, "Bayesian estimation of time varying systems," March 2012, written material for the course held in Spring 2012. [Online]. Available: http://www.lce.hut.fi/~ssarkka/course_k2012/full_course_booklet_2012.pdf 6, 8

[19] D. Simon, *Optimal State Estimation, H_∞ and Nonlinear Approaches.* Wiley-Interscience, 2006. 6

[20] F. Auger, M. Hilairet, J. Guerrero, E. Monmasson, T. Orlowska-Kowalska, and S. Katsura, "Industrial applications of the Kalman filter: A review," *Industrial Electronics, IEEE Transactions on*, vol. 60, no. 12, pp. 5458–5471, 2013. 6

[21] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, *Estimation with Applications to Tracking and Navigation.* New York, NY, USA: John Wiley & Sons, Inc., 2002. 6, 31, 34, 35

[22] D. D. S. Santana, "Navegação terrestre usando unidade de medição inercial de baixo desempenho e fusão sensorial com filtro de Kalman adaptativo suavizado," Ph.D. dissertation, EPUSP, 2011. 6

[23] M. Grewal and A. Andrews, "Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]," *Control Systems, IEEE*, vol. 30, no. 3, pp. 69–78, 2010. 6

[24] A. Gelb and T. Staff, *Applied Optimal Estimation*, A. Gelb, Ed. M.I.T. Press, 1974. [Online]. Available: http://books.google.com.br/books?id=GyuxQgAACAAJ 6

[25] M. S. Grewal and A. P. Andrews, *Kalman Filtering Theory and Practice Using Matlab*, 3rd ed. John Wiley and Sons, 2008. 6

[26] A. Doucet, N. De Freitas, and N. Gordon, Eds., *Sequential Monte Carlo methods in practice*, 2001. 6, 28

[27] E. A. Wan and R. van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, 2000, pp. 153 –158. 7, 12

[28] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *Automatic Control, IEEE Transactions on*, vol. 45, no. 5, pp. 910–927, May 2000. 7, 8, 9

[29] Y. Wu, D. Hu, M. Wu, and X. Hu, "A numerical-integration perspective on Gaussian filters," *Signal Processing, IEEE Transactions on*, vol. 54, no. 8, pp. 2910–2921, 2006. 7, 9

[30] A. Sakai and Y. Kuroda, "Discriminatively trained unscented Kalman filter for mobile robot localization," *Journal of Advanced Research in Mechanical Engineering (JARME)*, vol. 1, no. 3, pp. 153–161, 2010. 13, 16, 18

[31] R. van der Merwe, A. Doucet, N. de Freitas, and E. A. Wan, "The unscented particle filter," in *NIPS*, 2000, pp. 584–590. 15

[32] P. Abbeel, A. Coates, M. Montemerlo, A. Y. Ng, and S. Thrun, "Discriminative training of Kalman filters." in *Robotics: Science and Systems*, S. Thrun, G. S. Sukhatme, and S. Schaal, Eds. The MIT Press, 2005, pp. 289–296. 15, 16, 18, 19, 21

[33] D. Barber, *Bayesian Reasoning and Machine Learning.* Cambridge University Press, 2012. 15

[34] K. P. Murphy, *Machine learning: a probabilistic perspective*, Cambridge, MA, 2012. 15

[35] J. Dunik, M. Simandl, and O. Straka, "Adaptive choice of scaling parameter in derivative-free local filters," in *Information Fusion (FUSION), 2010 13th Conference on*, 2010, pp. 1–8. 16, 19

[36] R. Turner and C. E. Rasmussen, "Model based learning of sigma points in unscented Kalman filtering," *Neurocomput.*, vol. 80, pp. 47–53, Mar. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2011.07.029 16, 17, 18, 29, 30, 32, 37

[37] O. Straka, J. Dunik, and M. Simandl, "Gaussian sum unscented Kalman filter with adaptive scaling parameters," in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, 2011, pp. 1–8. 17, 19

[38] R. Turner and C. E. Rasmussen, "Model based learning of sigma points in unscented Kalman filtering," in *Machine Learning for Signal Processing (MLSP '10)*, Kittila, Finland, August 2010, pp. 178–183. 17, 18, 19, 30

[39] A. I. J. Forrester, A. Sobester, and A. J. Keane, *Engineering Design via Surrogate Modelling - A Practical Guide.* Wiley, 2008. 20, 22, 25

[40] J. C. Spall, *Introduction to stochastic search and optimization: Estimation, simulation, and control.* John Wiley and Sons, 2003. 22

[41] R. Hooke and T. A. Jeeves, "" direct search" solution of numerical and statistical problems," *J. ACM*, vol. 8, no. 2, pp. 212–229, Apr. 1961. [Online]. Available: http://doi.acm.org/10.1145/321062.321069 22, 30

[42] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1st ed. Addison-Wesley Professional, Jan. 1989. 22, 27

[43] R. Douc and O. Cappe, "Comparison of resampling schemes for particle filtering," in *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, Sept 2005, pp. 64–69. 24, 28

[44] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, 1 2006. 25

[45] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. of Global Optimization*, vol. 13, no. 4, pp. 455–492, Dec. 1998. [Online]. Available: http://dx.doi.org/10.1023/A:1008306431147 26, 27

[46] E. Brochu, V. M. Cora, and N. de Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," Department of Computer Science, University of British Columbia, Tech. Rep. TR-2009-23, November 2009. 26

[47] D. D. Cox and S. John, "Sdo: A statistical method for global optimization," in *in Multidisciplinary Design Optimization: State-of-the-Art*, 1997, pp. 315–329. 26

[48] D. Cox and S. John, "A statistical method for global optimization," in *Systems, Man and Cybernetics, 1992., IEEE International Conference on*, Oct 1992, pp. 1241–1246 vol.2. 26

[49] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *ICML*, J. Furnkranz and T. Joachims, Eds. Omnipress, 2010, pp. 1015–1022. 27

[50] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983. [Online]. Available: http://www.sciencemag.org/content/220/4598/671.abstract 27

[51] Y. Cao, M. Brubaker, D. Fleet, and A. Hertzmann, "Efficient optimization for sparse Gaussian process regression," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015. 29

[52] J. Quiñonero Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *J. Mach. Learn. Res.*, vol. 6, pp. 1939–1959, Dec. 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1046920.1194909 29

[53] M. A. Taddy, H. K. H. Lee, G. A. Gray, and J. D. Griffin, "Bayesian guided pattern search for robust local optimization," *Technometrics*, vol. 51, no. 4, pp. 389–401, 2009. [Online]. Available: http://dx.doi.org/10.1198/TECH.2009.08007 30

[54] J. Ko, D. J. Klein, D. Fox, and D. Haehnel, "Gp-ukf: Unscented Kalman filters with Gaussian process prediction and observation models," in *Intelligent Robots and Systems, 2007 IEEE/RSJ International Conference on*, 2007. 30

[55] J. Ko and D. Fox, "Gp-bayesfilters: Bayesian filtering using Gaussian process prediction and observation models," *Auton. Robots*, vol. 27, no. 1, pp. 75–90, 2009. [Online]. Available: http://dx.doi.org/10.1007/s10514-009-9119-x 30

[56] M. Psiaki, "The blind tricyclist problem and a comparative study of nonlinear filters: A challenging benchmark for evaluating nonlinear estimation methods," *Control Systems, IEEE*, vol. 33, no. 3, pp. 40–54, June 2013. 31, 65, 71

[57] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *Automatic Control, IEEE Transactions on*, vol. 54, no. 6, pp. 1254–1269, 2009. 32

[58] X. Li and Z. Zhao, "Measuring estimator's credibility: Noncredibility index," in *Information Fusion, 2006 9th International Conference on*, July 2006, pp. 1–8. 32

[59] G. Kitagawa, "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996. [Online]. Available: http://dx.doi.org/10.2307/1390750 35

[60] M. L. Psiaki, "Blind tricyclist problem matlab functions, example input/output data file, and example use in ekf calculations." *Ithaca, New York: Cornell Univ. [Online]*, 2013. [Online]. Available: http://gps.mae.cornell.edu/blind_tricyclist_models_simulation_example.zip 65, 68