

Ricardo Alberto Reis Silva Rendall

Comparaç o de abordagens monoescala e multiescala baseadas em dados e modelos para a filtragem de sinais em linha

Projeto de Tese na  rea cient fica de Engenharia Qu mica, orientada pelo Senhor Professor Marco Paulo Seabra dos Reis apresentada ao Departamento de Engenharia Qu mica da Faculdade de Ci ncias e Tecnologia da Universidade de Coimbra

10 de Julho de 2013



UNIVERSIDADE DE COIMBRA

Ricardo Alberto Reis Silva Rendall

Comparação de abordagens monoescala e multiescala baseadas em dados e modelos para a filtragem de sinais em linha

Projeto de Tese na área científica de Engenharia Química, orientada pelo Senhor Professor Marco Paulo Seabra dos Reis apresentada ao Departamento de Engenharia Química da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Supervisor:

Prof. Dr. Marco Paulo Seabra dos Reis

Coimbra

2013



UNIVERSIDADE DE COIMBRA

Agradecimentos

Gostaria de deixar aqui um agradecimento a todas as pessoas que contribuíram para realização desta dissertação. Primeiramente, ao professor Marco Seabra pelo apoio, disponibilidade e orientação. Agradecer pelas várias dúvidas que esclareceu e pelo ativo contributo que prestou. Um obrigado ao Tiago Rato que muito me ajudou, respondendo prontamente a todos os *e-mails*. Gostaria de agradecer a todos os professores da Universidade de Coimbra e em especial aos professores do Departamento de Engenharia Química da Universidade de Coimbra que contribuíram na minha formação e no desenvolvimento das minhas competências.

Um grande obrigado para a minha família, pelo apoio incondicional e pelo esforço que fizeram para me proporcionar a oportunidade de realizar um curso superior. Em especial à minha mãe, Maria Augusta Wahnnon dos Reis, ao meu tio, Nelson Wahnnon dos Reis e ao meu pai, Eduardo Alberto Gomes de Pina Silva Rendall, pelos valores e educação que me transmitiram.

Por fim, gostaria de agradecer aos amigos, tanto os de longa data como aqueles que fui conhecendo ao longo do curso, pelos bons e maus momentos que tivemos e pelo apoio e a minha namorada, pela sua compreensão e dedicação.

Seria impossível citar todos os nomes das pessoas que me ajudaram na realização desta dissertação e por isso, deixo aqui um muitíssimo obrigado a todos aqueles que de alguma forma contribuíram.

Resumo

Os filtros digitais são ferramentas que têm por objetivo remover o ruído presente num sinal contaminado, produzindo estimativas onde a contribuição do ruído é minimizada. O objetivo desta dissertação é a elaboração de um estudo comparativo envolvendo um número alargado de metodologias de filtragem, em diversos cenários de aplicação, permitindo obter linhas de orientação que auxiliem no processo da escolha de uma metodologia de filtragem.

O estudo comparativo divide-se em duas etapas, sendo que todos os sinais utilizados foram contaminados com 4 níveis de ruído ($SNR=5$, 10, 15 e 25). Na primeira fase, os filtros são sintonizados, minimizando o erro de previsão para o instante seguinte, para que as estimativas produzidas contenham o mínimo de ruído. Com os filtros assim sintonizados, na segunda fase efetua-se a comparação entre as diversas metodologias de filtragem, acedendo ao valor real da variável medida.

A fase de comparação iniciou-se com a pré-seleção de 7 metodologias de filtragem baseadas em dados, são elas: o filtro da média da janela móvel, o filtro exponencial, o filtro de Butterworth, o filtro de Chebyshev, o filtro elíptico, o filtro multiescala *online* e o filtro de Holt. Desta pré-seleção, destacou-se o filtro exponencial que teve o melhor desempenho num maior número de ocasiões, seguido do filtro multiescala *online*. O filtro exponencial foi mais adequado a situações onde a energia do ruído era elevada ($SNR=5$ e 10) enquanto que o filtro multiescala *online* foi melhor para baixos níveis de ruído ($SNR=15$ e 25). Uma análise mais cuidada permitiu evidenciar algumas desvantagens do filtro exponencial, nomeadamente os compromissos que este efetua quando o sinal contém fenómenos multiescala. A comparação dos filtros permitiu ainda concluir que a utilização do filtro de Holt ou do filtro de Chebyshev deve ser evitada na remoção de ruído branco.

O filtro exponencial e o filtro multiescala *online* foram de seguida comparados com o filtro de Kalman, uma metodologia que permite incorporar um modelo durante o processo de filtragem. O filtro de Kalman admite um modelo de espaço de estados e tem a vantagem de conseguir extrair informação relevante que se encontra a entrada do sistema. A sua aplicação removeu uma maior quantidade de ruído, quando comparado com as metodologias destacadas durante a pré-seleção e portanto, a sua utilização é recomendada.

A última metodologia de filtragem testada foi o filtro de Kalman multiescala. Os resultados deste filtro não foram os esperados, uma vez que seu desempenho foi inferior ao filtro exponencial e ao multiescala *online*. Uma vez que as matrizes do modelo foram estimadas, a utilização de um método de identificação de sistemas alternativo poderia conduzir a melhores desempenhos.

Abstract

Digital filters are tools that aim to remove noise present in a measured signal, producing estimates where the noise contribution is minimized. The objective of this thesis is the development of a comparative study involving a large number of filtering methods in several application scenarios, obtaining guidelines to assist in the process of choosing a filtering methodology.

The comparative study is divided in two phases and all the signals used were contaminated with 4 levels of noise ($SNR = 5, 10, 15$ and 25). In the first phase, the filters were tuned, minimizing the one-step-ahead prediction error, so that the estimates produced had low noise content. In the second stage, the comparison between different filters was made, accessing the real value of the measured variable.

The comparison began with the pre-selection of 7 data-driven methodologies: the moving average window, the exponential filter, the Butterworth filter, Chebyshev filter, the elliptic filter, the online multiscale filter and Holt's filter. This pre-selection highlighted the exponential filter which had the best performance in a greater number of occasions, followed by the online multiscale filter. The exponential filter was more suitable to situations where the presence of noise was high ($SNR = 5$ and 10) while the online multiscale filter was best for low levels of noise ($SNR = 15$ and 25). A more careful analysis showed the shortcomings of the exponential filter, including the tradeoffs that it makes when the signal has multiscale phenomena. The comparison also showed that the use of the Holt's filter or the Chebyshev filter for removing white noise should be avoided.

The exponential filter and the online multiscale filter were then compared with the Kalman filter, a methodology that allows incorporating a model during the filtering process. The Kalman filter admits a state space model and has the advantage of being able to extract relevant information at the input of the system. The implementation removed a greater amount of noise when compared to the methods outlined during pre-selection and therefore its use is recommended.

The last filtering methodology tested was the multiscale Kalman filter. The results of this filter were not as expected, since its performance was inferior to the exponential filter and the online multiscale filter. Since the matrices of the model were estimated using a system identification method, the possibility of using a different method could lead to better performance.

Índice

1 - Introdução	1
1.1 - Âmbito e motivação	1
1.2 - Objetivos.....	2
2 - Introdução aos filtros digitais.....	3
3 - Revisão bibliográfica	5
4 - Métodos.....	7
4.1 - Metodologias de filtragem sem recurso a modelos	7
4.1.1 - Filtro da média da janela móvel (JM).....	7
4.1.2 - Filtro exponencial (EWMA).....	8
4.1.3 - Filtro de Holt (FH).....	8
4.1.4 - Filtros: Butterworth (BWT), Chebyshev (CBS) e elíptico (ELI)	9
4.1.5 - Filtro multiescala <i>online</i> (OLMS).....	11
4.2 - Metodologias de filtragem que recorrem a modelos	15
4.2.1 - Filtro de Kalman (KF)	15
4.2.2 - Filtro de Kalman multiescala (MS-KF)	18
5 - Resultados	20
5.1 - Otimização dos parâmetros ajustáveis dos filtros.....	20
5.1.1 - Critérios de erro minimizados durante a otimização	20
5.1.2 - Sinais estudados e níveis de ruído utilizados.....	22
5.1.3 - Alguns resultados decorrentes da otimização dos parâmetros ajustáveis dos filtros	24
5.2 - Resultados do teste dos filtros	25
5.2.1 - Descrição da metodologia de comparação baseada no teste de hipóteses	26
5.2.2 - Resultados obtidos no teste dos vários filtros estudados	27
6 - Discussão dos resultados.....	34
6.1 - Discussão dos métodos de filtragem baseados em dados.....	34
6.2 - Discussão dos resultados obtidos da comparação entre o filtro exponencial, o filtro multiescala <i>online</i> e o filtro de Kalman	37
6.3 - Discussão dos resultados obtidos com o filtro exponencial, o de Kalman multiescala e o multiescala <i>online</i>	38
7 - Conclusões e perspectivas futuras.....	40
Bibliografia.....	42

Anexo I - Resultados da otimização dos parâmetros dos filtros.....	44
Anexo II - Resultados do teste dos filtros.....	52
Anexo III - Códigos em Matlab utilizados na otimização dos filtros.....	62
Anexo IV - Códigos em Matlab utilizados no teste dos filtros	77

Índice de Tabelas

Tabela 1 - Otimização dos filtros para o sinal de HeaviSine com SNR= 10 e 10 iterações....	24
Tabela 2 - Otimização dos filtros para o sinal de Doppler com SNR= 15 e 500 iterações.	25
Tabela 3 - Resultados do teste dos filtros sem modelos utilizando o sinal de HeaviSine com SNR=5 e 20 iterações.	28
Tabela 4 - Resultados do teste dos filtros sem modelos utilizando o sinal de blocos com SNR=15 e 20 iterações.	28
Tabela 5 - Otimização dos filtros para o sinal de <i>HeaviSine</i> com SNR=5 e 10 iterações.....	44
Tabela 6 - Otimização dos filtros para o sinal de <i>HeaviSine</i> com SNR=15 e 10 iterações.....	44
Tabela 7 - Otimização dos filtros para o sinal de <i>HeaviSine</i> com SNR=25 e 10 iterações.....	44
Tabela 8 - Otimização dos filtros para o sinal de <i>Doppler</i> com SNR=5 e 500 iterações.....	45
Tabela 9 - Otimização dos filtros para o sinal de <i>Doppler</i> com SNR=10 e 500 iterações.....	45
Tabela 10 - Otimização dos filtros para o sinal de <i>Doppler</i> com SNR=25 e 40 iterações.....	45
Tabela 11 - Otimização dos filtros para o sinal de <i>Blocks</i> com SNR=5 e 250 iterações.	45
Tabela 12 - Otimização dos filtros para o sinal de <i>Blocks</i> com SNR=10 e 250 iterações.	46
Tabela 13 - Otimização dos filtros para o sinal de <i>Blocks</i> com SNR=15 e 250 iterações.	46
Tabela 14 - Otimização dos filtros para o sinal de <i>Blocks</i> com SNR=25 e 20 iterações.	46
Tabela 15 - Otimização dos filtros para o sinal de <i>Bumps</i> com SNR=5 e 100 iterações.	46
Tabela 16 - Otimização dos filtros para o sinal de <i>Bumps</i> com SNR=10 e 100 iterações.	47
Tabela 17 - Otimização dos filtros para o sinal de <i>Bumps</i> com SNR=15 e 100 iterações.	47
Tabela 18 - Otimização dos filtros para o sinal de <i>Bumps</i> com SNR=25 e 20 iterações.	47
Tabela 19 - Otimização dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem, com SNR=5 e 40 iterações.	47
Tabela 20 - Otimização dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem, com SNR=10 e 40 iterações.	48
Tabela 21 - Otimização dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem, com SNR=15 e 40 iterações.	48
Tabela 22 - Otimização dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem, com SNR=25 e 20 iterações.	48
Tabela 23 - Otimização dos filtros para o sinal resultante do sistema com <i>overshoot</i> e resposta inversa com SNR=5 e 10 iterações para EWMA e OLMS e uma iteração para KF...48	48
Tabela 24 - Otimização dos filtros para o sinal resultante do sistema com <i>overshoot</i> e resposta inversa com SNR=10 e 10 iterações para EWMA e OLMS e uma iteração para KF.49	49
Tabela 25 - Otimização dos filtros para o sinal resultante do sistema com <i>overshoot</i> e resposta inversa com SNR=15 e 20 iterações para EWMA e OLMS e uma iteração para KF.49	49
Tabela 26 - Otimização dos filtros para o sinal resultante do sistema com <i>overshoot</i> e resposta inversa com SNR=25 e 20 iterações para EWMA e OLMS e uma iteração para KF.49	49
Tabela 27 - Otimização do filtro de Kalman para o sinal resultante da junção de dois sistemas de primeira ordem com SNR=5 e uma iteração.	49
Tabela 28 - Otimização do filtro de Kalman para o sinal resultante da junção de dois sistemas de primeira ordem com SNR=10 e uma iteração.	49

Tabela 29 - Otimização do filtro de Kalman para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=15$ e uma iteração.	49
Tabela 30 - Otimização do filtro de Kalman para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=25$ e uma iteração.	49
Tabela 31 - Resultados da otimização do filtro de Kalman multiescala.	50
Tabela 32 - Otimização do parâmetro dos filtros para o reator não isotérmico, com um $SNR=5$, 10 iterações para EWMA e OLMS e uma para MS-KF.	50
Tabela 33 - Otimização do parâmetro dos filtros para o reator não isotérmico, com um $SNR=10$, 10 iterações para EWMA e OLMS e uma para MS-KF.	50
Tabela 34 - Otimização do parâmetro dos filtros para o reator não isotérmico, com um $SNR=15$, 10 iterações para EWMA e OLMS e uma para MS-KF.	51
Tabela 35 - Otimização do parâmetro dos filtros para o reator não isotérmico, com um $SNR=15$, 10 iterações para EWMA e OLMS e uma para MS-KF.	51
Tabela 36 - Resultados do teste dos filtros para o sinal de <i>HeaviSine</i> com $SNR=10$ e 20 iterações.	52
Tabela 37 - Resultados do teste dos filtros para o sinal de <i>HeaviSine</i> com $SNR=15$ e 20 iterações.	52
Tabela 38 - Resultados do teste dos filtros para o sinal de <i>HeaviSine</i> com $SNR=25$ e 20 iterações.	53
Tabela 39 - Resultados do teste dos filtros para o sinal de <i>Doppler</i> com $SNR=5$ e 150 iterações.	53
Tabela 40 - Resultados do teste dos filtros para o sinal de <i>Doppler</i> com $SNR=10$ e 40 iterações.	53
Tabela 41 - Resultados do teste dos filtros para o sinal de <i>Doppler</i> com $SNR=15$ e 40 iterações.	54
Tabela 42 - Resultados do teste dos filtros para o sinal de <i>Doppler</i> com $SNR=25$ e 20 iterações.	54
Tabela 43 - Resultados do teste dos filtros para o sinal de <i>Blocks</i> com $SNR=5$ e 40 iterações.	54
Tabela 44 - Resultados do teste dos filtros para o sinal de <i>Blocks</i> com $SNR=10$ e 20 iterações.	55
Tabela 45 - Resultados do teste dos filtros para o sinal de <i>Blocks</i> com $SNR=15$ e 20 iterações.	55
Tabela 46 - Resultados do teste dos filtros para o sinal de <i>Blocks</i> com $SNR=25$ e 150 iterações.	55
Tabela 47 - Resultados do teste dos filtros para o sinal de <i>Bumps</i> com $SNR=5$ e 20 iterações.	56
Tabela 48 - Resultados do teste dos filtros para o sinal de <i>Bumps</i> com $SNR=10$ e 20 iterações.	56
Tabela 49 - Resultados do teste dos filtros para o sinal de <i>Bumps</i> com $SNR=15$ e 20 iterações.	56
Tabela 50 - Resultados do teste dos filtros para o sinal de <i>Bumps</i> com $SNR=25$ e 20 iterações.	57
Tabela 51 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=5$ e 20 iterações.	57

Tabela 52 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=10$ e 20 iterações.	57
Tabela 53 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=15$ e 20 iterações.	58
Tabela 54 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=25$ e 20 iterações.	58
Tabela 55 - Resultados do teste dos filtros para o sistema com resposta inversa e <i>overshoot</i> com $SNR=5$ e 10 iterações.	58
Tabela 56 - Resultados do teste dos filtros para o sistema com resposta inversa e <i>overshoot</i> com $SNR=10$ e 10 iterações.	59
Tabela 57 - Resultados do teste dos filtros para o sistema com resposta inversa e <i>overshoot</i> com $SNR=25$ e 10 iterações.	59
Tabela 58 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=5$ e 10 iterações.	59
Tabela 59 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=10$ e 10 iterações.	59
Tabela 60 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=15$ e 10 iterações.	59
Tabela 61 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=25$ e 10 iterações.	60
Tabela 62 - Resultados do teste dos filtros para o reator não isotérmico, com um $SNR=5$ e 10 iterações.	60
Tabela 63 - Resultados do teste dos filtros para o reator não isotérmico, com um $SNR=10$ e 10 iterações.	60
Tabela 64 - Resultados do teste dos filtros para o reator não isotérmico, com um $SNR=15$ e 10 iterações.	60
Tabela 65 - Resultados do teste dos filtros para o reator não isotérmico, com um $SNR=25$ e 10 iterações.	61

Índice de Figuras

Figura 1 - Resposta no domínio da frequência dos filtros: Butterworth, Chebyshev e elíptico	10
Figura 2 - Caixas de Heisenberg para filtros monoescala ((a) - no domínio frequência; (b) - no domínio tempo) e multiescala (c).	11
Figura 3 – Esquema da transformada de onduleta para um nível de decomposição 2 e para a transformada de Haar. Apresentam-se, a cinzento, os coeficientes necessários à reconstrução completa do sinal medido original (representado na base do esquema).....	12
Figura 4 - Dimensão da janela diádica ao longo da filtragem multiescala <i>online</i>	13
Figura 5 - Aplicação do filtro de Kalman. Os coeficientes sobre os quais o modelo é estimado estão a preto, sendo que a janela desloca-se quando uma nova medição se encontra disponível (de (a) para (b))......	19
Figura 6 - Sinais, com e sem ruído, utilizados no estudo comparativo. (a) corresponde ao sinal <i>HeavinSine</i> com um SNR=25, (b) <i>Doppler</i> com SNR=5, (c) <i>Blocks</i> com SNR=15, (d) <i>Bumps</i> com SNR=25, (e) é o sinal autorregressivo utilizado no conjunto de sistemas de primeira ordem e (f) é a sua resposta com SNR=10.....	23
Figura 7 - Sinais sem ruído, medidos e filtrados. (a) corresponde à resposta da junção dos dois sistemas de primeira ordem, com SNR=10 e filtrado com o filtro da janela móvel <i>online</i> utilizando o melhor parâmetro identificado na otimização. (b) corresponde ao sinal de <i>HeaviSine</i> com SNR=5, filtrado com o filtro de Butterworth, utilizando o melhor parâmetro identificado na otimização.....	25
Figura 8 - Sinais sem ruído, medidos e filtrados. (a) corresponde ao sinal de Doppler com SNR=5 filtrado com o filtro exponencial; (b) corresponde aos blocos, com SNR=15, filtrado com o filtro multiescala <i>online</i> ; (c) é a resposta da junção dos sistemas de primeira ordem com SNR=15, filtrado com o filtro de Holt; (d) corresponde aos <i>Bumps</i> , com SNR=5 e filtrado com o filtro de Butterworth.....	29
Figura 9 - Número de vezes que cada filtro sem modelo apresentou o menor erro (número de vitórias).....	30
Figura 10 - Número de vezes que cada filtro sem modelo apresentou o maior erro (número de derrotas).....	30
Figura 11 - Número de vitórias obtidas na filtragem da junção de dois sistemas de primeira ordem e na filtragem do sistema com <i>overshoot</i>	31
Figura 12 - Número de derrotas obtidas na filtragem da junção de dois sistemas de primeira ordem e na filtragem do sistema com <i>overshoot</i>	31
Figura 13 - Sinais sem ruído, medidos e filtrados decorrentes da simulação de um sistema com <i>overshoot</i> , com SNR=5 e 10 iterações. O sinal filtrado em (a) é obtido pela aplicação do filtro de Kalman e em (b) pela aplicação do filtro exponencial.	32
Figura 14 - Número de vitórias obtidas na filtragem dos sinais sem modelo (<i>HeaviSine</i> , <i>Doppler</i> , <i>Bumps</i> e <i>Blocks</i>) e na filtragem do sinal obtido pela junção de dois sistemas de primeira ordem.	32

Figura 15 - Número de derrotas obtidas na filtragem dos sinais sem modelo (<i>HeaviSine</i> , <i>Doppler</i> , <i>Bumps</i> e <i>Blocks</i>) e na filtragem do sinal obtido pela junção de dois sistemas de primeira ordem.	33
Figura 16 - Número de vitórias dos filtros exponencial e multiescala <i>online</i> para SNR=5 e 10.	35
Figura 17 - Número de derrotas dos filtros exponencial e multiescala <i>online</i> para SNR=15 e 25.	35
Figura 18 - Sinal com características multiescala filtrado com o filtro exponencial (a) e com o filtro multiescala <i>online</i> (b). O intervalo apresentado em (c) e (d) destaca as diferenças entre os filtros.	35
Figura 19 - Número de vitórias tendo em conta apenas o filtro da média da janela móvel, o de Butterworth e o elíptico.	36
Figura 20 - Número de vitórias obtidas pelo filtro exponencial, pelo de Kalman e pelo multiescala <i>online</i> . A filtragem utilizando o filtro de Kalman não teve em conta as entradas ao sistema.	37
Figura 21 - Número de vitórias obtidas na filtragem do reator não isotérmico.	39
Figura 22 - Número de derrotas obtidas na filtragem do reator não isotérmico.	39

1 - Introdução

Neste capítulo, introduz-se o âmbito do tema da presente dissertação, a filtragem de sinais. Adicionalmente apresenta-se a motivação, algumas aplicações de relevo, os objetivos e a organização do documento.

1.1 -Âmbito e motivação

O processamento de dados digitais é uma área que tem contribuído ativamente para o progresso tecnológico observado na indústria, particularmente desde os anos 1980. Neste contexto, a filtragem de dados tem um papel importante em diversas aplicações, constituindo um requisito de pré-processamento utilizado virtualmente em todos os domínios que requeiram a utilização de sinais (sensores, imagens, etc.) recolhidos do mundo exterior, como acontece na indústria, comunicações, medicina, entre outros.

Mais especificamente, entende-se por filtragem de dados a atividade que visa estimar o verdadeiro valor das variáveis medidas no instante atual, num conjunto de sinais corrompidos com ruído, usando dados disponíveis até ao presente. Pretende-se assim remover componentes indesejados num dado sinal, em particular aqueles de natureza estocástica (uma vez que componentes de natureza determinística devem ser corrigidos por outras metodologias complementares, como a calibração e correção de desvios sistemáticos).

No contexto da Engenharia Química, os filtros digitais estão presentes em todas as tarefas de controlo, monitorização, retificação e reconciliação de dados e otimização *online* ou *offline* de processos. Inerente a estas atividades, está a necessidade de obter medições precisas e exatas, que permitam conhecer em tempo útil o estado do sistema e todas as grandezas que o caracterizam. Como exemplo, os filtros digitais ao mitigarem o ruído presente nas medições, evitam que os sistemas de controlo respondam a alterações puramente aleatórias. Impede-se assim que esta variabilidade estocástica inerente aos sistemas de medição se propague e amplifique, com prejuízo para eficiência dos processos, sua segurança e qualidade do produto final. Assim sendo, a simples utilização de um filtro digital adequado terá um impacto direto e tangível na variabilidade final dos produtos, essencial para o estabelecimento de padrões de Qualidade elevados de produção, bem como na eficiência e segurança dos processos. Adicionalmente, contribui ainda para o aumento da vida útil dos elementos finais de controlo e atenua eventos de ressonância dos sistemas, de entre outras vantagens. A implementação de filtros digitais constitui assim uma mais-valia, com um claro retorno a vários níveis, sendo pois importante que seja conduzida da forma mais adequada a cada caso.

Dada a sua importância, existiu e existe um grande interesse no desenvolvimento de filtros cada vez mais eficazes, que abordam o mesmo problema, mas sob diferentes perspectivas. Tais estudos conduziram a uma bibliografia extensa que contempla um elevado número de metodologias de filtragem. Uma classificação possível, consiste em diferenciar entre metodologias que apenas têm em consideração os valores medidos durante a operação (baseadas em dados, *data-driven*) e metodologias que, para além das medições, têm em conta um modelo que descreve o processo, combinando estas duas fontes de informação (baseadas em modelos, *model-based*).

A necessidade de escolher o filtro mais adequado para uma dada aplicação constitui a principal motivação da presente dissertação. O gestor de processos, normalmente um Engenheiro de produção, é frequentemente confrontado com uma vasta gama de opções e soluções para este problema, no contexto das quais terá de tomar uma decisão que, não é de todo trivial. As comparações existentes na literatura são frequentemente pouco (ou mesmo muito pouco) exaustivas e muitas vezes o desempenho de uma nova metodologia de filtragem é apenas comparado com uma metodologia de natureza semelhante.

1.2 - Objetivos

O objetivo desta dissertação é a elaboração de um estudo comparativo envolvendo um número alargado de metodologias de filtragem, em diversos cenários de aplicação (tipos de sinais e níveis de ruído), com base no qual se irão extrair linhas de orientação para a seleção deste tipo de métodos em contextos práticos. Adicionalmente, testam-se, pela primeira vez, novas versões de metodologias multiescala e exploram-se as suas características mediante a análise dos resultados de filtragem. Do estudo comparativo resultará também uma análise custo-benefício entre as abordagens mais elaboradas e que requerem mais conhecimento *a priori* e os filtros mais simples e fáceis de sintonizar e implementar.

Tendo em conta os objetivos da dissertação, torna-se pertinente dar a conhecer a teoria subjacente aos filtros e de que forma estes são implementados. Assim, no próximo capítulo é apresentada uma breve introdução ao problema de filtragem e alguns conceitos básicos sobre filtros digitais. No capítulo 3 efetua-se uma revisão da bibliografia neste domínio, identificando os filtros com maior interesse para o presente trabalho e as áreas de investigação atualmente pertinentes. O capítulo 4 descreve a teoria que está na origem dos filtros selecionados. O desempenho dos filtros está dependente do valor de alguns parâmetros, os quais são analisados e otimizados no capítulo 5, recorrendo a dados de treino. Adicionalmente, algumas questões práticas da sua utilização serão também esclarecidas neste capítulo. De seguida, procede-se à determinação do desempenho dos filtros nos conjuntos de dados de teste. No capítulo 6 apresenta-se uma discussão sobre os resultados obtidos e algumas linhas orientadoras na escolha de um filtro. Por fim, o capítulo 7 resume as principais conclusões deste trabalho e perspectiva possíveis linhas de investigação e desenvolvimento nas quais ele poderá ter continuidade.

2 - Introdução aos filtros digitais

Os filtros digitais operam sobre sinais contaminados por ruído. O modelo adotado para o sinal medido (observado) é representado matematicamente pela equação (1).

$$x_t = x_t^r + e_t \quad (1)$$

Na equação (1), o sinal medido num dado instante (x_t) é obtido pela adição de duas componentes. A primeira corresponde à contribuição dada pelo sinal real (x_t^r) e representa o valor verdadeiro da variável medida. A segunda componente é o ruído (e_t), que corrompe a medição e não permite aceder ao valor real da variável que se deseja conhecer. Ao longo da dissertação considerar-se-á que o ruído segue uma distribuição normal, possuindo média nula e uma dada variância (σ_e), e portanto o termo do ruído da equação (1): $e \sim N(0, \sigma_e)$. Este ruído é denominado por ruído branco porque o seu espectro de frequências é constituído por todas as frequências com igual potência, ao contrário de um ruído colorido, onde a distribuição de potência não é equitativa pelas diferentes bandas de frequência. O ruído branco é um dos tipos de ruído mais encontrados na prática, resultando muitas vezes da interação entre vários efeitos (e.g. interferências eletromagnéticas, influência aleatória de variáveis não controladas como a temperatura externa e a humidade, etc.) cuja combinação segue uma lei normal, o que pode ser justificado pelo teorema do limite central.

O filtro digital tem por objetivo mitigar a contribuição da componente associada ao ruído, fornecendo uma estimativa que se aproxima o mais possível do valor real. Neste sentido, um filtro é entendido como um sistema com entradas e saídas, como indicado na equação (2).

$$\hat{x}_t = f(\{x_{t-i}\}_{i=0,1,2,\dots,I}, \{\hat{x}_{t-k}\}_{k=1,2,\dots,K}) \quad (2)$$

Como variável de saída do filtro, tem-se uma estimativa (\hat{x}_t) do valor real no tempo atual, que é uma função de variáveis de entrada. As entradas correspondem a medições efetuadas no instante t e/ou medições anteriores (x_t). Podem-se ter também como entradas, as estimativas produzidas pelo filtro em instantes anteriores (\hat{x}_{t-k}).

Uma primeira classificação que se pode adotar para distinguir os vários tipos de filtros digitais, consiste na distinção entre filtros *online* e *offline*. Filtros *online* seguem a descrição apresentada na equação (2) e são de grande importância quando uma estimativa da variável medida deve ser produzida imediatamente após a medição. A utilização de filtros *offline* é denominada por *smoothing* (alisamento) e o problema de estimação é fundamentalmente distinto, na medida em que se pode ter agora acesso a valores futuros das medições para cada instante em consideração (não é sequer correto chamar a este tipo de métodos de “filtros”, por esta razão). O objetivo é o mesmo mas neste caso a estimativa fornecida por estes filtros baseia-se no conhecimento das medições antes e após o instante t ($i = -I, \dots, 0, 1, \dots, J$). Um terceiro problema de estimação, distinto da filtragem e alisamento, é a previsão. Nesta, a estimativa a obter é referente ao futuro ($\{\hat{x}_{t+k}\}_{k=1,2,\dots,K}$), com base em dados recolhidos do presente (tempo t) ou passado. De maior

interesse prático destaca-se a previsão um passo a frente ($k = 1$), frequentemente usada em várias aplicações industriais, económicas e financeiras.

A presente dissertação incide sobre filtros *online*, uma vez que são estes os mais utilizados em processos industriais, permitindo a tomada de decisões em tempo útil e a condução de atividades de controlo de processos de forma mais efetiva.

Por outro lado, como já referido anteriormente, os filtros digitais podem também ser classificados como recorrendo a modelos ou somente a dados. Os filtros baseados em dados (*data driven*) apenas têm em conta a variável de saída do processo, i.e., a medição ou observação. As metodologias que recorrem a modelos (*model based*) carecem da disponibilidade de um modelo que relaciona as variáveis de entrada e de saída do processo (ou só as de saída, se o modelo não possuir variáveis de entrada), combinando a estimativa dada pelo modelo e a medição efetuada, por forma a obter uma estimativa mais adequada.

Outra classificação importante é relativa à linearidade. Filtros que obedecem à equação (3) são denominados de filtros lineares.

$$f(ax_1 + bx_2) = af(x_1) + bf(x_2) \quad (3)$$

onde a e b são constantes. Como se pode observar, uma combinação linear das entradas produz também uma combinação linear de saídas. De particular interesse são os sistemas (filtros) lineares invariantes no tempo, LTI. Neste caso, para além de serem lineares possuem a propriedade de que se a entrada for transladada por um dado período de tempo, a saída também sofrerá uma translação temporal da mesma magnitude. Filtros que não respeitem este princípio, designado por princípio da sobreposição, são chamados de filtros não-lineares.

Por fim, outra classificação possível diz respeito a filtros a uma só escala e filtros multiescala. Filtros a uma só escala são filtros que centram o processamento no domínio do tempo ou no domínio da frequência, a uma só escala. Eles possuem assim desempenho ótimo quando os sinais apresentam características na escala em que foram desenvolvidos. Os filtros multiescala, por outro lado, conduzem a sua ação em várias escalas temporais e de frequência em paralelo, possuindo por isso uma flexibilidade superior em se adaptar a sinais com características variadas e padrões multiescala.

3 - Revisão bibliográfica

Neste capítulo, analisa-se sumariamente a bibliografia disponível sobre filtros digitais, e os seus desenvolvimentos mais recentes. Como já foi referido, a bibliografia existente é extensa. No entanto alguns livros e artigos destacam-se tanto na qualidade com que introduzem o tema, como na forma como apresentam os vários desenvolvimentos. Textos introdutórios (Lyons 2001, Oppenheim *et al* 1997) permitem compreender os filtros e o seu funcionamento, dando particular relevância ao domínio da frequência, onde a transformada de Fourier surge como uma ferramenta indispensável na análise do espectro de um sinal. A análise neste domínio originou filtros como os designados por Butterworth, Chebyshev, elíptico (Smith, 1997), Notch (Karu 2001), e a transformada de Fourier numa janela deslizante (Krakovsky 2006), entre outros. Estes filtros têm por objetivo a remoção de bandas de frequências indesejadas que estejam a contaminar o sinal. Todos os filtros mencionados serão tidos em conta na fase da comparação, à exceção do filtro de Notch e da transformada de Fourier aplicada numa janela deslizante. O filtro de Notch remove uma banda de frequências centradas numa dada frequência central e não se esperam bons resultados uma vez que o ruído branco estende-se a todas as frequências. Estes filtros são bastante aplicados no caso de um ruído que possui uma frequência conhecida, como é o caso do ruído que provém da rede elétrica, que oscila a cerca de 60 Hz. A transformada de Fourier numa janela deslizante apresenta problemas de fronteira. O filtro consiste na aplicação da transformada de Fourier numa janela que se move à medida que uma nova medição fica disponível. No entanto, a estimativa mais recente produzida pelo filtro é precisamente a menos exata e portanto a metodologia é mais adequado à filtragem *offline* (Lyons 2001).

Da análise do domínio do tempo surgem filtros como o da média da janela móvel, o filtro exponencial (Reis *et al* 2009), o filtro de Holt (Gelper *et al* 2010) e o filtro de Holt-Winters (Chatfield *et al* 1988), entre outros. Estes filtros têm a finalidade de filtrar sinais codificados na escala do tempo. A teoria e a implementação são mais simples, o que justifica a sua utilização alargada. Dos filtros citados, o de Holt-Winters não será considerado na fase de comparação. Tanto o filtro de Holt como o de Holt-Winters são referenciados como métodos de previsão, apesar de poderem ser utilizados em modo de filtragem *online*. Assim sendo, escolheu-se apenas o de Holt, uma vez que o de Holt-Winters introduz somente algumas alterações no intuito de melhorar a previsão.

As metodologias apresentadas anteriormente são todas monoescala. Contudo, os sinais são muitas vezes compostos por fenómenos que ocorrem a várias escalas, o que conduziu ao desenvolvimento de filtros multiescala. Neste contexto, a transformada de onduletas constitui uma ferramenta essencial na determinação do conteúdo multiescala de um sinal. Esta transformada teve a sua origem no trabalho de Alfred Haar, por volta de 1910, que descobriu a primeira base ortogonal de onduletas. Muito tempo após a sua descoberta, diversos avanços se verificaram, desde a formulação de um método sistemático para gerar bases ortogonais (Daubechies 1988), até ao desenvolvimento da transformada rápida de onduleta (Mallat 1989). Estes desenvolvimentos

permitiram a construção de filtros multiescala *online* (Nounou *et al* 1999), com resultados bastante promissores.

Até agora só foram referidos filtros monoescala ou multiescala que não recorriam a um modelo do processo (*data driven*). No entanto, existindo um tal conhecimento disponível *a priori*, é de toda a vantagem que seja utilizado em conjunto com as medições experimentais. Em 1940, um primeiro avanço importante neste sentido foi dado com o aparecimento do filtro de Wiener (Brown *et al* 1997). Anos mais tarde, Kalman (Kalman 1960) publicou um artigo onde descreve uma metodologia de filtragem recursiva, melhorando vários aspetos do filtro de Wiener. Desde então várias versões do filtro de Kalman foram propostas, estendendo a sua aplicação a sistemas não lineares (Welch *et al* 2006). Esforços têm sido também dirigidos no sentido de identificar os seus parâmetros, nomeadamente as matrizes de variância-covariância associadas às várias componentes de ruído presentes no modelo de espaço de estados, necessário para implementação do filtro de Kalman (Akesson *et al* 2008). Outra metodologia que atualmente ganha muito relevo é a estimação de horizonte móvel (Haseltine *et al* 2005). Trata-se de um método de filtragem que utiliza modelos, mas que está mais direcionado a sistemas não lineares utilizando rotinas de otimização para maximizar a probabilidade *a posteriori* partindo de um conjunto de medições. Esta metodologia sai fora do conjunto de métodos aqui considerados, pelo facto de, para feitos de comparação, o filtro de Kalman congrega os aspetos essenciais a considerar, dada a natureza linear dos sistemas usados no estudo de comparativo.

Mais recentemente foram propostos filtros que combinam um modelo e a decomposição multiescala do sinal, analisando modelos a múltiplas escalas (Bakshi 1999, Reis 2009). Esta metodologia permite combinar o filtro de Kalman com a filtragem multiescala *online*, utilizando um modelo nas escalas onde o sinal está concentrado, e portanto onde o ruído é menos preponderante. Esta abordagem é bastante promissora uma vez que combina a estimativa ótima produzida pelo filtro de Kalman com a componente multiescala do sinal. Muita da investigação que se realiza presentemente está relacionada com a presença de *outliers*, no sentido de aumentar a robustez dos filtros (Sardy *et al* 2001, Doymaz *et al* 2001, Padonou *et al* 2013). Uma vez que os dados aqui simulados não contêm *outliers*, estas metodologias saem também fora do âmbito do presente estudo.

Mais filtros estão disponíveis na literatura. Em particular: Reis *et al* (2009), Nounou *et al* (1999) e Smith (1997) são referências onde se apresentam e discutem metodologias tanto *online* como *offline*, a uma só escala e multiescala, com e sem modelos.

4 - Métodos

Neste capítulo, introduz-se a teoria que suporta os filtros anteriormente referidos e que serão alvo de comparação neste estudo, nomeadamente: o filtro da média da janela móvel, o filtro exponencial, o filtro de Holt, o filtro Butterworth, o filtro Chebyshev, o filtro elíptico, o filtro multiescala *online*, o filtro de Kalman e, por fim, o filtro de Kalman multiescala. Estes filtros constituem um conjunto alargado de metodologias de filtragem e cobrem quase todo o espetro de tipos de filtros *online* disponíveis.

4.1 - Metodologias de filtragem sem recurso a modelos

Inicialmente apresentar-se-á o conjunto de filtros que não recorrem a modelos, i.e., filtros que têm apenas em consideração as medições efetuadas da variável de saída do processo.

4.1.1 - Filtro da média da janela móvel (JM)

O filtro da média da janela móvel é um dos filtros mais utilizados devido a sua simplicidade e facilidade de implementação. Este filtro estima o valor presente, \hat{x}_t , através do cálculo da média aritmética de uma janela de dimensão m (equação (4)).

$$\hat{x}_t = \frac{1}{m+1} \sum_0^m x_{t-m} \quad (4)$$

Quando uma nova medição se torna disponível, a janela move-se, incorporando a nova medição e retirando a medição mais antiga. A motivação do filtro da média da janela móvel reside no pressuposto de que em medições suficientemente próximas, o sinal é sensivelmente constante e as flutuações que ocorrem são aleatórias, e assim podem ser atenuadas pelo uso da média. A escolha da dimensão da janela (parâmetro m) é um compromisso entre sinais muito ruidosos (m pequeno), mas que respondem rapidamente a alterações no sinal real, e sinais muito suaves (m grande), que apesar de não terem quase nenhum ruído remanescente, podem ser demasiado enviesados, não permitindo detetar ou detetando tardiamente variações rápidas do sinal real (Reis *et al* 2009). A média móvel é um dos filtros preferidos para a filtragem de sinais codificados no domínio do tempo, e portanto, pouco interessante no domínio das frequências, sendo pouco eficiente na separação de bandas de frequências (Smith 1997). Este filtro é um exemplo de um filtro de resposta finita (FIR).

Esta classificação surge uma vez que cada estimativa dada pelo filtro é calculada com base num número finito de medições, número esse normalmente muito menor que o número de medições disponíveis, e a estabilidade do filtro é garantida (Lyons 2001). Neste contexto, estabilidade significa que, se por algum motivo, as medições passarem a ser nulas, a estimativa dada pelo filtro também será nula. Neste caso tal sucederá após um número de pontos igual à dimensão da janela.

4.1.2 - Filtro exponencial (EWMA)

O filtro exponencial, como o próprio nome indica, estima o valor real do sinal, atribuindo um peso (α) ao valor medido no presente, sendo que o peso das medições passadas decai exponencialmente. Computacionalmente é implementado de acordo com a equação (5), que constitui a sua forma recursiva.

$$\hat{x}_t = \alpha x_t + (1 - \alpha)\hat{x}_{t-1} \quad (5)$$

O parâmetro α varia entre 0 e 1. Uma vez mais, este parâmetro tem de ser otimizado visto que valores elevados produzem sinais ruidosos e valores baixos tendem a suavizar em demasia, distorcendo as características do sinal real. Quando expandido, o filtro exponencial é expresso por um somatório que engloba todas as medições (equação (6)).

$$\hat{x}_t = \sum_{i=0}^{\infty} \alpha(1 - \alpha)^i x_{t-i} \quad (6)$$

Pela equação (6), constata-se o decaimento exponencial, que constitui uma das características deste filtro. Pontos mais recentes têm um maior peso enquanto pontos mais distantes possuem um peso menor, o que permite uma melhor adaptação a alterações no sinal real, ao contrário do que acontece na janela móvel, onde todos os pontos possuem igual importância. Este filtro é classificado como um filtro de resposta infinita (IIR) ou recursivo uma vez que cada estimativa produzida pelo filtro é calculada com base em todas as medições efetuadas. A estabilidade depende do parâmetro α e não é garantida, podendo originar uma resposta não nula muito tempo depois das medições o serem.

4.1.3 - Filtro de Holt (FH)

O filtro de Holt (Gelper *et al* 2010) é de certa forma uma extensão do filtro exponencial, aplicado a séries temporais que possuem uma tendência. A tendência é entendida como uma

variação da média, para o que se propõe uma nova variável com o intuito de a monitorar. A sua implementação passa pela utilização das equações (7) e (8).

$$\hat{x}_t = \lambda_1 x_t + (1 - \lambda_1)(\hat{x}_{t-1} + F_{t-1}) \quad (7)$$

$$F_t = \lambda_2(\hat{x}_t - \hat{x}_{t-1}) + (1 - \lambda_2)F_{t-1} \quad (8)$$

A variável F_t corresponde à alteração da média e este filtro utiliza dois parâmetros, λ_1 e λ_2 , onde cada um pesa exponencialmente a medição e a alteração na média, respetivamente. O filtro é bastante interessante do ponto de vista da previsão, sendo esta a sua aplicação de maior relevo. Uma vez que captura a alteração da média, o procedimento do cálculo do valor futuro é também bastante intuitivo (equação 9),

$$\hat{x}_{t+k|t} = \hat{x}_t + kF_t \quad (9)$$

onde $\hat{x}_{t+k|t}$ é a previsão k passos à frente utilizando a informação disponível até ao tempo t . Pressupõe portanto que a variação que se verifica na média irá manter-se ao longo do período de previsão. Na filtragem *online*, k toma o valor 0, e o termo \hat{x}_t é entendido como um nível local, que se traduz na estimativa (Padonou, Roustant, & Lutz, 2013).

4.1.4 -Filtros: Butterworth (BWT), Chebyshev (CBS) e elíptico (ELI)

Os filtros Butterworth, Chebyshev e elíptico enquadram-se numa categoria diferente dos demais. Foram desenvolvidos e muito utilizados pelos Engenheiros Eletrotécnicos e abordam o problema de filtragem na escala de frequências, onde os sinais desejados e/ou o ruído oscilam com uma frequência conhecida. O objetivo passa então por identificar e recuperar/remover tais frequências.

A transformada de Fourier discreta (DFT) permite fazer a transposição de um sinal na escala do tempo para a escala de frequências, conduzindo ao seu espectro de frequência. Nesta nova escala, o sinal original é dado pela soma de sinusóides com diferentes frequências e magnitudes, as quais constituem uma base ortogonal para o espaço dos sinais em questão. A DFT também pode ser utilizada em filtragem, definindo-se um valor de corte para a magnitude dos coeficientes e anulando aqueles que possuam valores abaixo deste limite, considerando que estes correspondem a ruído. O sinal é depois reconstruído na escala original do tempo, com os coeficientes assim processados. Também existe a janela de DFT (WDFT) em que, à semelhança da média da janela móvel, uma janela deslizante percorre o sinal, efetuando a filtragem desejada. Como já foi referido, este método não é muito aconselhado para a filtragem *online*, uma vez que sofre de problemas de fronteira (Lyons 2001).

A transformada de Fourier pode ser também utilizada para conhecer a resposta de frequência de filtros do tipo FIR, como no caso da janela móvel, aplicando a transformada aos coeficientes do filtro. No entanto, filtros recursivos, como é o caso do filtro exponencial, só podem ser analisados

pela transformada de Z, que é a versão discreta da transformada de Laplace. Noções como polos e zeros são transpostos para o domínio discreto, sendo que neste caso um sistema discreto é estável quando os seus polos se encontram dentro de um círculo de raio unitário no diagrama Z (a componente imaginária de Z em função da componente real).

A construção dos filtros Butterworth, Chebyshev e elíptico pode ser efetuada segundo três tipos de métodos: o método da invariância ao impulso, o método bilinear e métodos de otimização. Para mais detalhes, o leitor pode consultar a literatura (Lyons 2001, Smith 1997). Sendo o *Matlab* o *software* utilizado para a construção dos filtros, nele é implementado o método bilinear. Em suma, define-se o número de polos e a resposta desejada através de uma função transferência contínua e estável $G(s)$, e de seguida a variável s é substituída de acordo com a equação (10),

$$s = \frac{2}{t_s} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) \quad (10)$$

onde t_s é o tempo de amostragem. A equação (10) permite transpor do domínio contínuo para o domínio discreto os polos da função transferência, garantindo que estes se irão encontrar no interior de um círculo de raio unitário. Tendo obtido a função transferência no domínio discreto, $G(Z)$, aplica-se a transformada inversa em Z, originando um conjunto de coeficientes que constituem o filtro no domínio do tempo. A aplicação do filtro passa pela determinação de um conjunto de coeficientes que atuam sobre as medições e sobre estimativas passadas produzidas pelo filtro, conforme a equação (11).

$$\hat{x}_t = \sum_{k=0}^N b_k \times x_{t-i} + \sum_{k=1}^M a_k \times \hat{x}_{t-k} \quad (11)$$

A questão prende-se agora na diferença entre os três filtros. A Figura 1 ilustra a resposta no domínio da frequência destes três filtros, a qual permite uma melhor compreensão das suas diferenças.

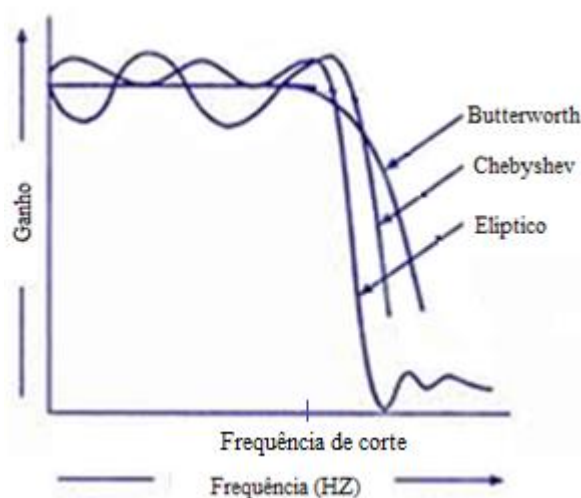


Figura 1 - Resposta no domínio da frequência dos filtros: Butterworth, Chebyshev e elíptico.

Como se pode observar na Figura 1 os filtros atenuam frequências acima de uma frequência de corte. O filtro de Butterworth não apresenta oscilações na zona abaixo da frequência de corte, e por isso é denominado de completamente plano. No que se refere ao filtro Chebyshev, este apresenta oscilações na zona inferior à frequência de corte, mas permite que a zona de transição seja mais curta, o que poderá melhorar o seu desempenho em determinadas aplicações. Por fim, o filtro elíptico conduz a oscilações tanto acima como abaixo da frequência de corte, mas faz com que a zona de transição diminua ainda mais. A origem destas diferenças prende-se com a posição dos polos dos três filtros. No caso do filtro Butterworth, os polos encontram-se num círculo, enquanto que no caso do filtros de Chebyshev e elíptico, os polos encontram-se numa elipse, dispostos de forma diferente consoante o filtro (Karu 2001). De notar ainda que a diferença analisada na Figura 1 corresponde a 3 filtros com o mesmo número de polos, e que o desempenho pode ser melhorado pela adição de mais polos. Tanto no filtro Chebyshev como no elíptico, torna-se necessário definir a percentagem de oscilação, sendo que existem valores recomendados (Smith 1997, MATLAB 7.14 2012). Considerou-se ainda que cada filtro era composto por 6 coeficientes (3 polos e 3 zeros), um número suficientemente elevado para garantir um bom desempenho, evitando que a construção dos filtros se torne muito complicada.

4.1.5 - Filtro multiescala *online* (OLMS)

Até agora foram apresentados apenas filtros lineares e que abordam o problema de filtragem na escala de frequências ou na escala do tempo e impõem, implicitamente, um compromisso entre a exatidão com que representam alterações localizadas no tempo e a capacidade para remover o ruído. No entanto, os sinais reais apresentam características diversas, agregando fenómenos localizados no tempo mas também dinâmicas que são bem descritas por uma análise de frequências. Surge então a necessidade de recorrer a métodos multiescala, com recurso à teoria das onduletas. Estas representam uma família de funções com capacidade de reconstruir sinais reais de uma forma muito eficiente, sendo por isso amplamente utilizadas na compressão de dados, no reconhecimento, amplificação e filtragem de sinais sonoros, no estudo de fenómenos sísmicos, entre outras aplicações (Graps 1995). O conceito multiescala é melhor entendido recorrendo às caixas de incerteza de Heisenberg (Figura 2).

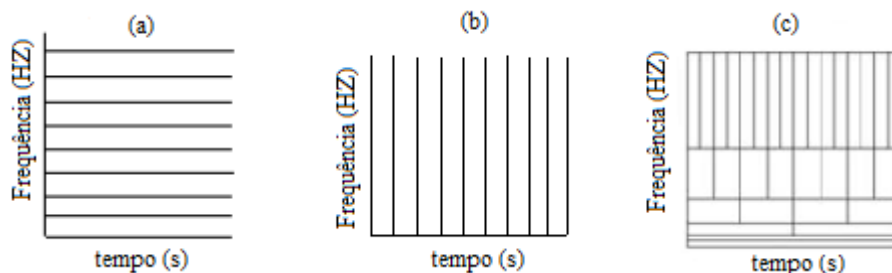


Figura 2 - Caixas de Heisenberg para filtros monoescala ((a) - no domínio frequência; (b) - no domínio tempo) e multiescala (c).

Métodos

A Figura 2 permite de forma mais clara evidenciar as vantagens da abordagem multiescala. Enquanto que filtros monoescala conseguem descrever adequadamente fenômenos localizados no domínio frequência ou tempo, mas não nos dois simultaneamente, os filtros multiescala possuem uma resolução bastante interessante nas duas escalas, através de uma divisão adequada do domínio tempo×frequência: para frequências mais elevadas, a que correspondem picos e transições abruptas, o filtro multiescala possui boa resolução no tempo, sendo esta a escala característica deste tipo de fenômenos; para frequências mais baixas, a que correspondem tendências e oscilações suaves, o filtro possui boa resolução na frequência, uma vez que é aí que reside a forma mais adequada para as descrever.

A transformada de onduleta pode ser obtida através da convolução do sinal com um filtro passa-baixo, H e um filtro passa-alto, G . A passagem pelo filtro H é equivalente a projetar o sinal num espaço funcional de funções de aproximação (cuja base é formada pelas chamadas funções escala). A passagem pelo filtro G equivale a projetar o sinal no espaço funcional complementar, de funções de detalhe (cuja base é composta pelas chamadas funções onduleta). As equações (15) e (16) representam a operação matricial análoga às respectivas operações de convolução (somente para efeitos de exposição, pois a multiplicação matricial é muito menos eficiente devido a esparcidade das matrizes H e G).

$$a_j = H a_{j-1} \quad (15)$$

$$d_j = G a_{j-1} \quad (16)$$

Resumidamente, a transformada de onduleta conduz a um conjunto de coeficientes de aproximação (a_j) e de detalhe (d_j), onde o índice J indica a escala de análise. A escala varia entre 0 (que corresponde ao sinal original) até a escala mais elevada (L), correspondente às frequências mais baixas (Figura 3). A reconstrução do sinal original é conseguida através da aplicação da transformada inversa de onduleta, retendo todos os coeficientes de detalhe a todas as escalas (de 1 a L) e os coeficientes de aproximação na escala mais elevada.

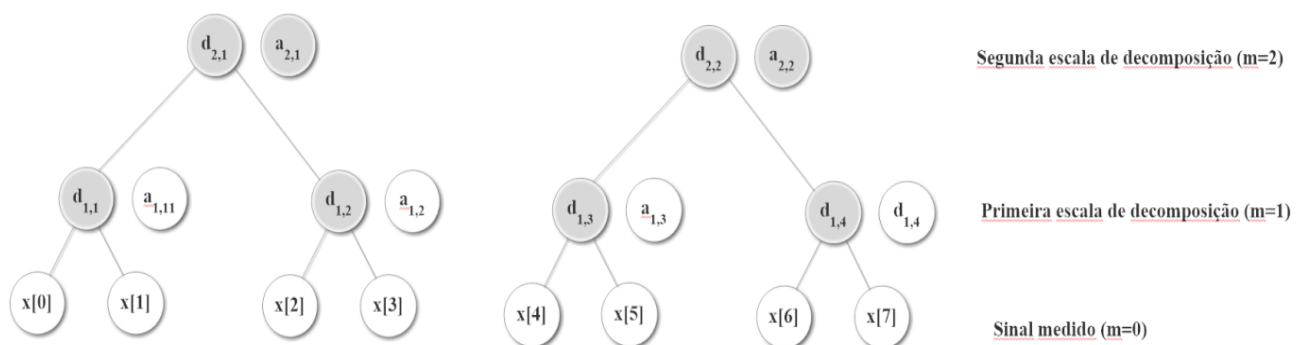


Figura 3 – Esquema da transformada de onduleta para um nível de decomposição 2 e para a transformada de Haar. Apresentam-se, a cinzento, os coeficientes necessários à reconstrução completa do sinal medido original (representado na base do esquema).

A filtragem recorrendo à transformada de onduleta baseia-se no facto de que características dominantes do sinal são descritas por um pequeno número de coeficientes, sendo que o ruído se dispersa por todos os coeficientes, com uma magnitude relativamente reduzida. Estes últimos podem assim ser eliminados, escolhendo adequadamente um valor de corte. Até 1999, a aplicação da transformada de onduleta estava limitada ao uso *offline*, contudo foi proposta (Nounou *et al* 1999) uma metodologia para adaptar o seu uso ao contexto *online*, denominada por filtragem multiescala *online* (OLMS). Esta técnica consiste em 4 passos fundamentais:

- 1) Decompor o sinal numa janela de dimensão diádica;
- 2) Calcular o valor de corte a cada escala, mantendo os coeficientes acima deste valor e reconstruir o sinal retificado na escala do tempo;
- 3) Reter o último ponto da janela;
- 4) Quando uma nova medição se torna disponível, a janela desloca-se para abranger este novo ponto, mantendo a maior dimensão diádica possível. Esta fase é melhor entendida recorrendo a Figura 4. Como se pode observar, quando 4 medições estão disponíveis, essas 4 são utilizadas na filtragem. A dimensão da janela permanece em 4 e desloca-se à medida que novas medições são efetuadas. A partir do momento em que 8 medições se tornem disponíveis, a dimensão da janela passa a ser 8 até que sejam feitas 16 medições e assim sucessivamente.

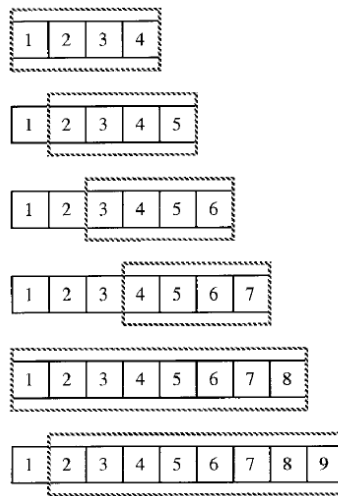


Figura 4 - Dimensão da janela diádica ao longo da filtragem multiescala *online*.

Existem duas formas de efetuar a operação de *thresholding* com valor de corte (t_m): *hard thresholding* e *soft thresholding*. No *hard thresholding*, os coeficientes abaixo do valor de corte são simplesmente eliminados (o seu valor é reduzido a 0). No *soft thresholding*, para além de eliminar aqueles que são inferiores ao valor de corte, também se comprimem os restantes, por um valor igual ao valor de corte (equação (17)), tendo como fundamento o facto de que os coeficientes de magnitude mais elevada também estão contaminados pelo ruído.

$$\tilde{d}_m = \begin{cases} \text{sign}(d_m)(|d_m| - t_m), & |d_m| > t_m \\ 0, & |d_m| < t_m \end{cases} \quad (17)$$

Na equação 17, \tilde{d}_m representa os coeficientes de onduleta processados. Na abordagem de filtragem multiescala *online*, adotou-se o *soft thresholding* uma vez que este normalmente conduz a uma melhor qualidade visual dos dados filtrados e também apresenta menos artefactos (Nounou e Bakshi 1999). Artefactos são perturbações que ocorrem no sinal reconstruído quando uma variação abrupta no sinal não coincide com uma posição diádica, i.e., quando a função utilizada para descrever essa variação e a variação em si não se alinham. A determinação do valor de corte é feita recorrendo ao método proposto por Donoho (Bakshi 1999). O valor de corte a cada escala é dado pela equação (18).

$$t_m = \sigma_m \sqrt{2 \log N} \quad (18)$$

Onde σ_m é o desvio padrão dos erros à escala em consideração, o qual pode ser por vezes determinado pelo conhecimento da natureza do erro. No entanto, em diversas situações esta informação é desconhecida e o desvio padrão é estimado de forma robusta, de acordo com a equação (19).

$$\sigma_m = \frac{1}{0.6745} \text{mediana}(|d_m|) \quad (19)$$

Por fim, uma das decisões mais importantes prende-se com a escolha do tipo de onduletas, uma vez que existe uma ampla variedade de famílias, sendo as de Haar e Daubechies as mais conhecidas (na verdade, as onduletas de Haar são um caso particular das onduletas de Daubechies, correspondendo à sua família com um menor número de momentos anuladores). Optou-se pelas de Haar uma vez que os respetivos filtros de passa-alto e passa-baixo são causais, i.e., não necessitam de conhecer medições futuras para estimar o valor atual. Existe também uma família de onduletas denominada de onduletas com fronteiras corrigidas (*bounded corrected*), que tentam minimizar os problemas que surgem nas fronteiras da transformada de onduleta, à semelhança do que ocorre com WDFT. No entanto, a sua performance nem sempre é garantidamente melhor que a de Haar, como pode ser constatado da análise dos resultados obtidos por Nounou *et al* (1999), optando-se por isso por esta última.

Como para os restantes filtros, também aqui é necessária a otimização dos parâmetros de sintonização, particularmente no que diz respeito à escala de decomposição. Um valor baixo para a profundidade de decomposição (J pequeno) conduz à presença de bastante ruído no sinal aproximado (a) e, por isso, mesmo aplicando o valor de corte, o sinal filtrado ainda iria conter uma contribuição significativa de ruído. Por outro lado, a consideração de uma escala muito elevada para a decomposição, (J elevado), conduz a uma dispersão excessiva da energia do sinal verdadeiro pelas diversas escalas, confundindo-se com o ruído, o que levará à distorção do sinal original após aplicação da operação de *thresholding*.

4.2 - Metodologias de filtragem que recorrem a modelos

Tendo-se apresentado as bases teóricas relativas aos filtros que não recorrem a modelos, consideram-se agora metodologias que têm a capacidade de incorporar, de forma ótima, um modelo do processo na operação filtragem. Em particular, refere-se o filtro de Kalman, dada a sua importância neste contexto e ubiquidade em aplicações industriais desta natureza.

4.2.1 - Filtro de Kalman (KF)

Os filtros descritos anteriormente requerem apenas os sinais obtidos do processo, não requerendo ou permitindo sequer incorporar no processo de filtragem qualquer informação adicional do sistema em estudo. No entanto, muitas vezes dispõe-se de informação da dinâmica do sistema, sob a forma de um modelo matemático (mecânico ou empírico) e pretende-se conciliar o modelo e a medição por forma a obter uma melhor estimativa do valor atual da variável medida. Em 1940, Wiener considerou o problema de filtragem para o caso em que o espectro do sinal e do ruído se sobrepunham, quando se está na posse de um modelo do sistema. A solução encontrada pondera todos os valores medidos por forma a obter uma melhor estimativa do estado atual, minimizando o erro quadrático médio entre a estimativa e o sinal real. No entanto, esta abordagem possui algumas limitações, como por exemplo: as necessidades de memória e velocidade de processamento que crescem à medida que mais amostras são recolhidas. O tratamento de sistemas com múltiplas entradas e saídas também se torna mais difícil.

O filtro de Kalman surge então como uma resposta a estas limitações. O seu carácter recursivo e portanto muito eficiente (comparativamente ao filtro de Winer), e a utilização de notação matricial na linguagem de modelação, são avanços que lhe conferiram destaque na filtragem de dados. O artigo que descreve o filtro foi publicado em 1960 (Kalman, 1960), tendo já sido utilizado em inúmeras aplicações, principalmente na área de navegação autónoma ou assistida, mas também em processos industriais.

O primeiro requisito do filtro é que o sistema seja descrito por um modelo de espaço de estados (equação (20)). Têm-se também de conhecer a relação entre as variáveis medidas e o estado do sistema (equação (21)).

$$\theta_t = A\theta_{t-1} + Bu_{t-1} + w_{t-1} \quad (20)$$

$$x_t = C\theta_t + v_t \quad (21)$$

As variáveis w_t e v_t representam o ruído do processo e da medição, respetivamente. Assume-se que são independentes, têm médias nulas e variância Q e R , respetivamente. Define-se então uma estimativa *a priori* do estado, $\hat{\theta}_t^-$ (equação (22)) e uma estimativa *a posteriori*, $\hat{\theta}_t$.

$$\hat{\theta}_t^- = A\hat{\theta}_{t-1} + Bu_{t-1} \quad (22)$$

Esta estimativa *a priori* prevê o estado do sistema no instante t , tirando partido da informação disponível em $t-1$. No instante t , a medição x_t fica disponível e o objetivo é combinar esta medição e a estimativa *a priori*, obtendo a estimativa *a posteriori* (equação (23)), que minimize o erro quadrático médio.

$$\hat{\theta}_t = \hat{\theta}_t^- + K_t (x_t - C\hat{\theta}_t^-) \quad (23)$$

em que K_t é o ganho de Kalman. De seguida, definem-se dois erros, um *a priori* (equação (24)) e outro *a posteriori* (equação (25)) e as suas respetivas variâncias (equação (26) e equação (27)).

$$e_t^- = \theta_t - \hat{\theta}_t^- \quad (24)$$

$$e_t = \theta_t - \hat{\theta}_t \quad (25)$$

$$P_t^- = E[e_t^- e_t^{-T}] \quad (26)$$

$$P_t = E[e_t e_t^T] \quad (27)$$

O ganho K_t é definido como aquele que minimiza a covariância do erro *a posteriori*, sendo calculado pela equação (28). Mais informação sobre o desenvolvimento do FK e outras expressões para o cálculo do ganho está disponível na literatura técnica dedicada a este tema (Brown *et al* 1997). Em suma, K_t é calculado aplicando as condições necessárias à função objetivo envolvendo o erro *a posteriori* e, por manipulação matemática, obtêm-se uma expressão que permite o seu cálculo.

$$K_t = P_t^- C^T (CP_t^- H^T + R)^{-1} \quad (28)$$

É de notar que a equação (28) representa um compromisso entre a incerteza do processo e a incerteza da medição. No caso em que a incerteza do processo é baixa (P_t^- tende para zero) o filtro dá maior importância ao valor previsto, $\hat{\theta}_t^-$, uma vez que o ganho tende para zero. No caso em que a incerteza da medição é menor, o filtro atribui maior peso a medição (Welch *et al* 2006).

A aplicação do filtro de Kalman depende da disponibilidade de um modelo. No entanto, modelos mecanísticos não estão disponíveis em muitos casos práticos, uma vez que a quantidade de fenómenos a modelar se torna elevada, e mesmo que se tenha um modelo, os parâmetros nele contido não são conhecidos com exatidão suficiente. Assim sendo, existe a necessidade de recorrer a modelos empíricos que, tendo em conta as entradas e saídas do sistema, permitem estimar os parâmetros do modelo (matrizes A, B, e C) e as variâncias dos ruídos (R e Q). Trata-se portanto de um problema de identificação de sistemas, tema já bem estabelecido na literatura (Ljung 1987). Em particular, um método bastante útil no presente contexto, pertencente à categoria dos métodos de identificação de subespaço de estados (*subspace identification methods*, SIM) é o denominado N4SID. Trata-se de um método baseado em subespaços desenvolvido em 1994 (Overschee *et al* 1994), o qual se baseia num modelo constituído pelas equações (20) e (21). A partir destas, pode-se escrever um modelo de espaço de estados estendido (equação (29)).

$$Y_f = \Gamma_f X_t + H_f U_f + G_f E_f \quad (29)$$

Em que o subscrito f indica um horizonte futuro de previsão, Γ_f é a matriz de observabilidade estendida, equação (30). H_f e G_f são matrizes com uma estrutura tipo Toeplitz (equação (31) e equação (32), respetivamente).

$$\Gamma_f = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{f-1} \end{bmatrix} \quad (30)$$

$$H_f = \begin{bmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{f-2}B & CA^{f-3}B & \dots & D \end{bmatrix} \quad (31)$$

$$G_f = \begin{bmatrix} I & 0 & \dots & 0 \\ CB & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{f-2}K & CA^{f-3}K & \dots & I \end{bmatrix} \quad (32)$$

A matriz U_f tem a forma de uma matriz tipo Hankel (equação (33)), à semelhança de Y_f e E_f . A matriz X_K é a sequência de estados (equação 34).

$$U_f = \begin{bmatrix} u_t & u_{t+1} & \dots & u_{t+N-1} \\ u_{t+1} & u_{t+2} & \dots & u_{t+N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{t+f-1} & u_{t+f} & \dots & u_{t+f+N-2} \end{bmatrix} \quad (33)$$

$$X_K = [\theta_t, \theta_{t+1}, \dots, \theta_{t+N-1}] \quad (34)$$

As equações (20) e (21) são rescritas de forma a obter uma relação entre o estado atual e os estados passados, em contraste com as equações anteriores que se centram num futuro de previsão. Desta forma, o termo X_t da equação (29) pode ser substituído por uma relação com valores das medições passadas. A equação (29) pode ser simplificada de acordo com alguns pressupostos e mediante manipulações algébricas é possível o cálculo das diversas matrizes do sistema. Conhecendo as matrizes, o ganho de Kalman pode ser calculado, sendo que o *software* Matlab (utilizando o pacote de identificação de sistemas) fornece-o diretamente.

Foi aqui apresentado, resumidamente, as principais etapas e algumas equações do método N4SID uma vez que a dedução de todas as expressões e uma análise mais aprofundada é longa, saindo fora do âmbito desta dissertação. O leitor interessado é direcionado para artigos bibliográficos da especialidade (Qin 2006, Favoreel *et al* 2000). As vantagens no uso do método N4SID são várias, nomeadamente a facilidade com que se adapta a sistemas multivariados de ordem elevada, sendo que a determinação dos parâmetros depende apenas da escolha da ordem do sistema que também pode ser estimado pela decomposição em valores singulares. O método é não iterativo, o que permite obter a solução com reduzido esforço computacional, sem problemas de garantia de convergência e mínimos locais.

A ordem do sistema constitui o parâmetro a otimizar para que o filtro produza melhores estimativas. Utilizar uma ordem baixa não permite capturar características presentes no sistema

enquanto que a especificação de uma ordem elevada pode conduzir a um sobreajuste aos dados, e o filtro perde a sua capacidade de filtragem.

Por fim, é importante referir que uma das mais-valias do filtro de Kalman prende-se com a capacidade de congregar toda a informação disponível sobre o modelo, variáveis de entrada e da saída. Todos os filtros anteriormente apresentados só levavam em conta a variável de saída do sistema, enquanto que o filtro de Kalman consegue incorporar a informação disponível sobre as entradas do sistemas, encontrando uma relação que permite prever a sua evolução e remover o ruído de forma adequada. Por conseguinte, espera-se que o seu desempenho seja muito bom (na verdade ótimo, se as suas condições forem satisfeitas), devido à conjugação de todos estes fatores.

4.2.2 - Filtro de Kalman multiescala (MS-KF)

O filtro de Kalman multiescala é um dos últimos avanços em termos de filtros digitais, combinando a transformada de onduleta com o filtro de Kalman. Trata-se portanto de um método multiescala baseado em modelos. A modelação multiescala é introduzida na literatura (Reis 2009, Bakshi 1999), onde diversos métodos (identificando modelos do tipo AR, PCA, PLS, etc.) podem ser utilizados a cada escala por forma a obter uma melhor estimativa dos coeficientes, particularmente nas escalas onde o sinal está concentrado. A aplicação do filtro passa pela decomposição do sinal de saída em diversas escalas, sendo que em cada uma delas um modelo de espaço de estados é estimado e o filtro de Kalman aplicado (Chui *et al* 2009).

Introduz-se neste trabalho uma nova versão do filtro de Kalman multiescala. Esta versão utiliza uma janela de dimensão diádica, mas não a maior possível por forma a eliminar problemas que possam surgir devido à necessidade excessiva de recursos computacionais. Assim sendo, após a definição de uma dada escala de decomposição (J_{dec}), a janela deve conter um número de pontos suficientes para que essa decomposição seja conseguida ($2^{J_{dec}}$). Nas escalas onde o sinal está concentrado (normalmente nas escalas mais elevadas), aplica-se o filtro de Kalman, melhorando a estimativa dos coeficientes relativos a estas escalas. Nas escalas mais contaminadas pelo ruído (normalmente escalas mais baixas), aplica-se a metodologia de *soft thresholding* utilizada no método de filtragem *online* multiescala. Esta metodologia é oportuna neste contexto, uma vez que nas escalas mais baixas o processo é quase inteiramente estocástico e sem autocorrelação, não sendo portanto vantajosa a utilização de um modelo do processo. Por outro lado, dispendo-se de mais coeficientes a estas escalas, a operação de *thresholding* pode ser também conduzida de uma forma mais consistente.

A implementação do filtro é ilustrada na Figura 5. Numa janela de dimensão diádica, aplica-se a transformada de onduleta, retendo o último coeficiente em cada escala. Quando uma nova medição fica disponível, a janela desloca-se de forma a incorporar essa nova medição, e descartando simultaneamente a medição mais antiga. Aplica-se então novamente a transformada de onduleta, retendo-se uma vez mais o último coeficiente a cada escala, e assim sucessivamente. Desta forma,

Métodos

constrói-se um conjunto formado pelos coeficientes a cada escala correspondentes aos tempos atuais, os quais são retificados posteriormente utilizando o filtro de Kalman, nas escalas onde este é aplicado, i.e., onde a estrutura dinâmica do sinal está concentrada. Em cada instante, são apenas necessários os últimos coeficientes a cada escala para reconstruir o sinal no domínio do tempo, para uma dada janela (coeficientes a preto na Figura 5).



Figura 5 - Aplicação do filtro de Kalman. Os coeficientes sobre os quais o modelo é estimado estão a preto, sendo que a janela desloca-se quando uma nova medição se encontra disponível (de (a) para (b)).

Existe também a possibilidade de fornecer ao filtro de Kalman multiescala informação referente às variáveis de entrada do sistema, quando estas estão disponíveis. Assim como o sinal de saída é decomposto, também se decompõem os sinais de entrada a diversas escalas, e o modelo a cada escala é estimado com base nos coeficientes dos sinais de entrada e saída nessa mesma escala.

A escala de decomposição é o parâmetro ajustável deste filtro. A semelhança do que acontecia com o filtro multiescala *online*, escalas muito baixas removem uma baixa quantidade de ruído enquanto que escalas elevadas podem conduzir a remoção de características importantes.

Pela análise anterior espera-se que este filtro tenha um desempenho bastante interessante, em sistemas complexos com características multiescala.

5 - Resultados

A utilização de qualquer um dos filtros descritos anteriormente é condicionada pela necessidade de especificação de determinados parâmetros que traduzem alguns dos compromissos mencionados anteriormente, nomeadamente entre a capacidade de adaptação a variações rápidas do sinal real e à sua capacidade de filtrar o ruído. Assim sendo, este capítulo é subdividido em duas secções. Na primeira secção, é efetuada a otimização dos filtros, onde cada filtro é sintonizado de forma a que as estimativas produzidas sejam o mais próximo possível do valor do sinal real. Neste sentido, é necessário definir quais os sinais a serem estudados e qual o critério utilizado para a otimização dos parâmetros ajustáveis dos filtros. Na segunda secção, os filtros assim otimizados são testados extensivamente recorrendo a dados de teste e os seus desempenhos comparados sistematicamente. Para simplificar a comparação, é inicialmente efetuada uma pré-seleção dos filtros pertencentes à categoria daqueles não recorrem a modelos (o grupo mais numeroso), apurando aqueles que apresentam os melhores desempenhos. Estes passarão então a uma segunda fase, onde serão comparados com os filtros que recorrem a modelos, cujo desempenho se espera superior. Esta estratégia permite evidenciar as vantagens e limitações dos vários filtros, mantendo o número de métodos em comparação num nível razoável e conduzindo a uma análise mais detalhada.

5.1 - Otimização dos parâmetros ajustáveis dos filtros

Neste subcapítulo, aborda-se o problema da sintonização dos filtros digitais. Apresentam-se os diferentes critérios de otimização usados, os tipos de sinais utilizados na comparação e a metodologia adotada para quantificar o ruído que perturba os sinais. Por fim, apresentam-se alguns dos resultados da otimização dos filtros.

5.1.1 - Critérios de erro minimizados durante a otimização

A otimização dos filtros pode ser feita recorrendo a diversos critérios de minimização dos erros (Kohler *et al* 2005), sendo que cada um possui as suas vantagens e desvantagens. O critério mais natural passaria pela escolha do parâmetro que minimize a diferença entre a estimativa produzida pelo filtro (\hat{x}_t) e o valor do sinal real (x_t^r), o qual, em estudos de simulação, é conhecido uma vez que os dados utilizados são gerados. Este critério, apesar de ser aquele que seguramente conduziria aos melhores resultados, distorce a realidade que os métodos visam abordar, na medida em que em situações práticas, não é possível aceder ao conhecimento integral do valor da variável

Resultados

medida. Outra abordagem passaria pela minimização da diferença entre o valor medido (x_t) e a estimativa dada pelo filtro, para o mesmo instante (\hat{x}_t). Este caso conduziria a filtros sem capacidade de filtragem, que simplesmente se limitariam a fornecer estimativas tão próximas quanto possíveis dos valores observados. Por exemplo, no caso da média móvel, esta tenderia a minimizar a dimensão da janela, uma vez que nessa situação a estimativa produzida seria próxima/igual ao valor medido, erroneamente desprezando os valores passados.

Considerando o uso que se pretende dar aos métodos em teste nesta dissertação, especificou-se como critério para obtenção do parâmetro de sintonização, $\hat{P}_{ótimo}$, aquele que conduziria à minimização do erro de previsão para o instante seguinte (*one-step-ahead predictor*), i.e., o parâmetro que minimize a diferença entre a previsão para o instante $t+1$ (utilizando a informação disponível em t) e o valor medido no instante $t+1$, de acordo com a equação (35).

$$\hat{P}_{ótimo} = \arg \min \left(\sqrt{\frac{\sum (\hat{x}_{t+1|t} - x_{t+1})^2}{N}} \right) \quad (35)$$

Em que N é o número de pontos. Este critério de minimização do erro permite obter estimativas adequadas, desde que os sinais disponíveis do processo tenham uma dimensão suficiente, com a vantagem de poder ser implementado no dia-a-dia, recorrendo ao histórico de medições.

Para a previsão correspondente ao instante seguinte ($\hat{x}_{t+1|t}$), e no caso do filtro de Holt, esta é feita explicitamente, supondo uma variação constante na média. No caso do filtro exponencial, prova-se (Gelper *et al* 2010) que a melhor previsão para o futuro é o próprio valor estimado no instante atual ($\hat{x}_{t+1|t} = \hat{x}_t$). No caso dos restantes filtros, esta mesma abordagem foi seguida para estimar os correspondentes valores futuros. Assim sendo, admitiu-se que, à semelhança do filtro exponencial, a melhor previsão corresponderia ao valor filtrado atual. No entanto, esta metodologia tem limitações, principalmente quando os sinais são de curta duração e/ou existe uma contribuição aleatória pouco apreciável entre amostras sucessivas. O número de pontos disponíveis deverá ser elevado, caso contrário esta metodologia sofreria das mesmas condicionantes que afetam o segundo critério de erro anteriormente mencionado, i.e., conduziria ao sobreajuste dos dados. Considerando, por exemplo, um sinal monotonamente crescente e em que o valor das medições é sucessivamente maior (baixa contribuição do ruído), o parâmetro que minimiza o erro entre a medição no instante seguinte e a previsão, corresponde à situação em que a previsão é o próprio valor medido, sem qualquer retificação. Isto ocorre pois o valor medido atual está mais próximo do valor que será observado futuramente e a inclusão de qualquer valor passado aumentaria o erro. Para este tipo de sinais, a aplicação preliminar de abordagens de correção de tendência é adequada.

A necessidade de uma elevada taxa de amostragem remete para o campo da amostragem de sinais, que estuda a influência da frequência de amostragem relativamente às frequências encontradas no sinal, de forma a evitar situações de *aliasing* (impossibilidade de distinguir diferentes frequências no sinal em estudo). Estes efeitos não são aqui considerados, uma vez que a frequência de amostragem depende da aplicação. Assume-se no entanto que a frequência de

Resultados

amostragem foi definida de forma a permitir observar as características dinâmicas relevantes dos sinais.

A otimização é realizada pelo método Monte Carlo. Em cada simulação, define-se um valor para o parâmetro de cada filtro (dimensão da janela para o filtro da média da janela móvel, α para o filtro exponencial, etc.) e de seguida efetua-se a filtragem. São calculados dois erros: a raiz quadrada da média da soma do quadrado dos erros entre o sinal real e o sinal retificado ($RMSE_r$, equação (36)) e a raiz quadrada da média da soma do quadrado dos erros de previsão, dados pela diferença entre o sinal observado e o sinal retificado ($RMSE_{ap}$, equação (37)). Após várias realizações, determinam-se as respetivas médias, \overline{RMSE}_r e \overline{RMSE}_{ap} , e o parâmetro com menor média do erro de previsão é escolhido como sendo o ótimo. Determina-se também o desvio padrão para cada caso: $\sigma_{\overline{RMSE}_r}$ e $\sigma_{\overline{RMSE}_{ap}}$. O parâmetro que minimiza \overline{RMSE}_r é denominado por parâmetro ótimo ideal ($P_{ótimo,r}$) e o que minimiza \overline{RMSE}_{ap} denomina-se parâmetro ótimo aparente ($\hat{P}_{ótimo}$). O desvio percentual entre os parâmetros é também determinado.

$$RMSE_r = \sqrt{\frac{\sum(\hat{x}_t - x_t^r)^2}{N}} \quad (36)$$

$$RMSE_{ap} = \sqrt{\frac{\sum(\hat{x}_{t+1|t} - x_{t+1})^2}{N}} \quad (37)$$

Outra questão relativamente à otimização dos parâmetros dos filtros, e que afeta também a comparação dos filtros, prende-se com a inicialização dos mesmos. Filtros como Butterworth ou Chebyshev, aquando da aquisição da primeira medição, necessitam de valores passados para produzir um valor retificado. Contudo, esse valor passado não está ainda, naturalmente, disponível. Para lidar com estas situações, adotou-se a abordagem mais simples que consiste em esperar até que um número suficiente de medições esteja disponível para produzir a primeira estimativa.

5.1.2 - Sinais estudados e níveis de ruído utilizados

Recordando o objetivo da dissertação: pretende-se avaliar o desempenho de diferentes filtros perante diversos sinais e diferentes níveis de ruído, de forma a proporcionar uma base de conhecimento para a seleção de filtros em contextos práticos futuros. Neste sentido primeiramente estabelece-se a noção de nível de ruído de um sinal e, de seguida, apresentam-se os sinais estudados aos quais se irá adicionar ruído. O nível de ruído num sinal pode ser quantificado pela razão entre a variância do sinal real e a variância do ruído (SNR), conforme a equação (36).

$$SNR = 10 \times \log\left(\frac{Var(x^r)}{Var(e)}\right) \quad (36)$$

Resultados

A equação (36) permite que, conhecendo a variância do sinal real, se possa calcular a variância que o ruído deve ter para que sinais diferentes sejam contaminados na mesma proporção.

Seguindo a prática na literatura especializada, estudar-se-ão primeiramente 5 tipos de sinais com características distintas. São eles: *HeaviSine*, *Doppler*, *Blocks* e *Bumps*. Estes sinais são habituais na literatura e representam, de uma forma isolada e simples, fenômenos muitas vezes encontrados na prática (Doymaz *et al* 2001). A este conjunto de sinais adiciona-se outro produzido pela junção de dois sistemas de primeira ordem, com constantes de tempo diferentes: 10 e 150, ambos com ganhos igual a 2. A entrada deste sistema corresponde à junção de um sinal autorregressivo (equação (37)) e alguns degraus de magnitude e ocorrência aleatórios. Os sinais são contaminados com 4 níveis de ruído, $SNR=5, 10, 15$ e 25 . A Figura 6 apresenta os sinais mencionados com os diversos níveis de ruído.

$$u_t = 3 + u_{t-1} + 0.3e_t \quad (37)$$

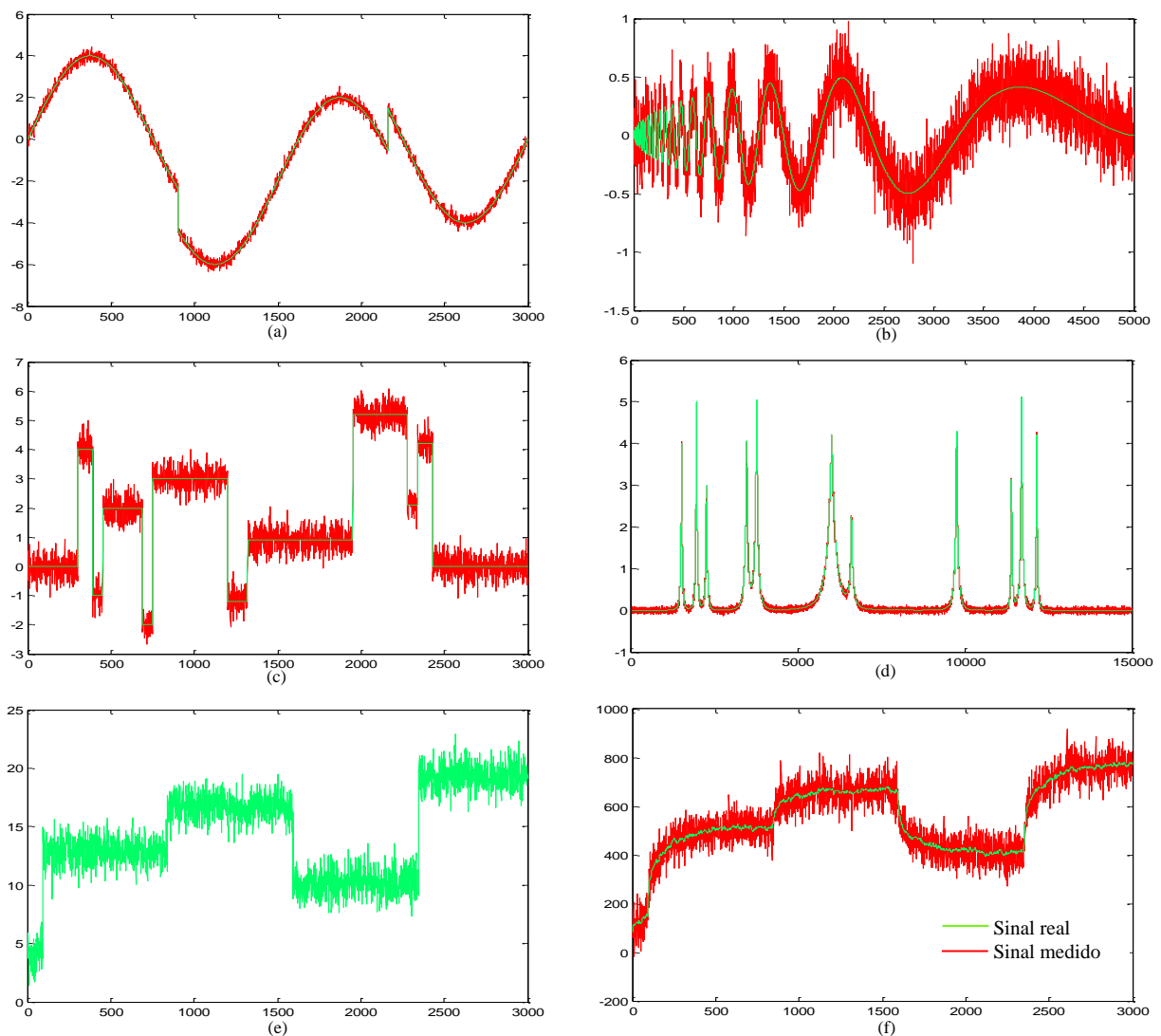


Figura 6 - Sinais, com e sem ruído, utilizados no estudo comparativo. (a) corresponde ao sinal *HeaviSine* com um $SNR=25$, (b) *Doppler* com $SNR=5$, (c) *Blocks* com $SNR=15$, (d) *Bumps* com $SNR=25$, (e) é o sinal autorregressivo utilizado no conjunto de sistemas de primeira ordem e (f) é a sua resposta com $SNR=10$.

5.1.3 - Alguns resultados decorrentes da otimização dos parâmetros ajustáveis dos filtros

A título ilustrativo, apresentam-se de seguida alguns dos resultados obtidos na otimização dos vários filtros que incorporam este estudo, para vários tipos de sinais. Uma vez que os diferentes sinais têm características distintas, a otimização deve ser feita caso-a-caso, para que todos os filtros se apresentem nas melhores condições para a executar a tarefa de filtragem. A Tabela 1 apresenta os resultados para a otimização dos filtros, após filtragem do sinal de *HeaviSine* com $SNR=10$. A Tabela 2 apresenta os resultados da otimização para o sinal de *Doppler* com $SNR=15$. A Figura 7 apresenta o sinal real, o medido e o filtrado após otimização dos parâmetros para os filtro da média da janela móvel e para o filtro de Butterworth, utilizando a resposta da junção dos sistemas de primeira ordem e o sinal de *HeaviSine*, respetivamente.

Os resultados completos para a otimização dos parâmetros para os vários filtros podem ser consultados no Anexo I. Como se pode observar pelos resultados, a metodologia de otimização é bastante eficaz, estimando parâmetros que se encontram, na maior parte dos casos, próximos do seu valor ideal ($P_{ótimo,r}$). Destaca-se ainda o facto da média do erro de previsão (desvio entre o valor estimado e o valor observado) ser sempre superior à média do erro real (desvio entre o valor estimado e o valor real), i.e., o filtro apresenta sempre um melhor desempenho do que aquele que é estimado na prática. Este fenómeno é justificado pela própria natureza da ação de filtragem. O filtro produz uma estimativa que se pretende que seja próxima do valor real, e por isso a diferença entre o valor retificado e o valor real é baixo visto que a contribuição do ruído é minimizada. Por outro lado, o cálculo do erro de previsão envolve diretamente a utilização do valor medido, o qual está contaminado com o ruído, inflacionando a respetiva medida do erro.

A implementação da estimação dos parâmetros (i.e., o “treino” dos filtros) foi efetuada na plataforma de cálculo Matlab, sendo o código utilizado apresentado no Anexo III.

Tabela 1 - Otimização dos filtros para o sinal de *HeaviSine* com $SNR=10$ e 10 iterações.

Filtros	$\hat{P}_{ótimo}$	$P_{ótimo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	19	20	5	0.9792	0.2686	4.24E-03	3.95E-03
EWMA	0.0942	0.0922	2.17	0.9776	0.2627	4.24E-03	3.89E-03
BWT	0.0482	0.0462	4.3	0.9816	0.2766	4.27E-03	4.21E-03
CBS	0.0322	0.0322	0	0.9945	0.3199	4.21E-03	4.21E-03
ELI	0.0322	0.0322	0	0.9941	0.3192	4.27E-03	4.08E-03
OLMS	5	5	0	0.9884	0.2937	4.40E-03	6.45E-03
HF	0.3307	0.2553	29.53	1.1542	0.5808	4.61E-02	6.71E-02
	0.1009	0.0918	9.91				

Nota: Para JM, o parâmetro a otimizar é a dimensão da janela (m), para o EWMA é o peso (α), para os filtros: BWT, CBS e ELI otimiza-se a frequência de corte (f_c), para o DWT a escala de decomposição (J) e para o HW os valores de λ_1 e λ_2 .

Resultados

Tabela 2 - Otimização dos filtros para o sinal de Doppler com SNR= 15 e 500 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	5	5	0	0.0601	0.027	2.68E-05	2.24E-05
EWMA	0.3367	0.3046	10.53	0.0597	0.0264	2.68E-05	2.24E-05
BWT	0.2305	0.2025	13.85	0.061	0.0287	3.13E-05	2.68E-05
CBS	0.1664	0.1464	13.69	0.0627	0.0324	2.68E-05	2.68E-05
ELI	0.1624	0.1424	14.07	0.0622	0.0315	2.68E-05	2.68E-05
OLMS	2	2	0	0.0614	0.0313	2.24E-05	2.24E-05
HF	0.99	0.99	0	1.384	1.1481	4.95E-02	4.16E-02
	0.1708	0.1508	13.26				

Nota: Para JM, o parâmetro a otimizar é a dimensão da janela (m), para o EWMA é o peso (α), para os filtros: BWT, CBS e ELI otimiza-se a frequência de corte (f_c), para o DWT a escala de decomposição (J) e para o HW os valores de λ_1 e λ_2 .

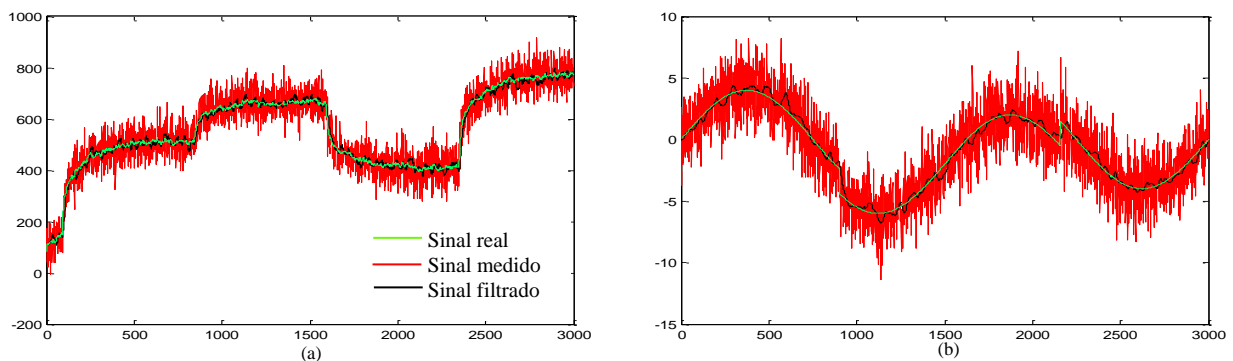


Figura 7 - Sinais sem ruído, medidos e filtrados. (a) corresponde à resposta da junção dos dois sistemas de primeira ordem, com SNR=10 e filtrado com o filtro da janela móvel *online* utilizando o melhor parâmetro identificado na otimização. (b) corresponde ao sinal de *HeaviSine* com SNR=5, filtrado com o filtro de Butterworth, utilizando o melhor parâmetro identificado na otimização.

5.2 - Resultados do teste dos filtros

A otimização dos filtros permitiu determinar os parâmetros mais adequados dos filtros para cada combinação de sinal e nível de ruído. Nesta secção, os filtros são utilizados na filtragem de novos sinais não utilizados no seu ajuste (dados de teste) e é realizado um teste de hipóteses para aceder à significância da diferença registada nos seus desempenhos. Numa primeira fase, apenas serão considerados os filtros que não utilizam modelos, uma vez que estes são em maior número e o seu desempenho é em princípio inferior aos filtros baseados em modelos. Estes últimos serão analisados numa segunda fase, em conjunto com os filtros com melhor desempenho na primeira fase. Faz-se assim uma pré-selecção que permite destacar os filtros baseados em dados com maior potencial e só depois efetuar a sua comparação com metodologias mais complexas.

5.2.1 -Descrição da metodologia de comparação baseada no teste de hipóteses

O teste de hipóteses é uma ferramenta estatística que permite inferir sobre a veracidade de um conjunto de hipóteses, i.e., permite rejeitar ou não uma afirmação acerca de um dado parâmetro (e.g. média) de uma população. Existem testes que permitem inferir sobre a média de uma população, sobre a sua variância, entre outros parâmetros populacionais (Montgomery *et al* 2003). Neste caso, o interesse está na diferença entre as médias de duas populações: deseja-se saber se, em média, um filtro produz melhores resultados que outro, i.e., se existe evidência suficiente que confirme que os erros produzidos por um filtro são significativamente inferiores aos erros produzidos por outro, ou se a diferença observada é apenas fruto do acaso. É de notar que, contrariamente ao procedimento utilizado durante a otimização, considerar-se-á como medida do erro a diferença entre o valor filtrado e o valor real, i.e., $RMSE_r$, uma vez que só assim é possível determinar, sem qualquer ambiguidade, o desempenho de cada filtro, tirando partido do facto de todos os dados serem simulados. Por outras palavras, acede-se desta forma ao verdadeiro desempenho dos métodos de filtragem, o que só é possível por se tratar de um estudo baseado em simulações computacionais.

O teste de hipóteses é aplicado aos resultados obtidos por cada par distinto de filtros nas várias réplicas dos sinais que ambos processaram. Cada um destes métodos, quando aplicados a uma dado cenário simulado, é caracterizado por possuir uma dada população de erros quadráticos médios a eles associadas, as quais são caracterizadas por possuir médias populacionais, $\mu_{RMSE,1}$ e $\mu_{RMSE,2}$, e respetivos desvios padrão. Assim, os filtros ao serem utilizados para retificar várias realizações do mesmo tipo de sinal conduzem a um conjunto de erros. A partir destes, é possível estimar os parâmetros populacionais em causa, a partir dos respetivos estimadores amostrais, $\overline{RMSE}_{r,1}$ e $\overline{RMSE}_{r,2}$, ou diretamente a sua diferença, através de um teste-t emparelhado. A partir destes estimadores não enviesados e da respetiva variabilidade pode-se então aferir qual o que possui melhor desempenho. Esta é a ideia base, a qual é detalhada mais especificamente de seguida.

O código em Matlab utilizado no cálculo dos erros foi construído para que em cada iteração todos os filtros processassem o mesmo sinal (Anexo IV), permitindo assim implementar um teste-t para amostras emparelhadas, o qual tem maior potência para discriminar entre os desempenhos dos métodos, que o correspondente teste-t para amostras independentes (Montgomery *et al* 2003). O teste-t para amostra emparelhadas consiste em decidir se o seguinte parâmetro populacional dado pela diferença das duas médias populacionais em comparação,

$$\mu_D = \mu_{RMSE,1} - \mu_{RMSE,2} \quad (40)$$

é igual ou diferente de zero (equações (41) e (42)):

$$H_0: \mu_D = 0 \quad (41)$$

$$H_1: \mu_D \neq 0 \quad (42)$$

Resultados

Para cada realização do sinal, os erros obtidos ($RMSE_r$) pelos filtros são emparelhados, calculando-se a diferença entre eles, d_j . Calcula-se a média destas diferenças, \bar{d} , e determina-se a estatística t_0 (equação (42)).

$$t_0 = \frac{\bar{d}}{s_D/\sqrt{n}} \quad (42)$$

Em que s_D é o desvio padrão da amostra das diferenças e n é o número de pares. Definem-se dois valores críticos e a hipótese nula é rejeitada quando a estatística se encontra na chamada zona crítica (equação (43)).

$$t_0 > t_{\alpha/2, n-1} \text{ ou } t_0 < -t_{\alpha/2, n-1} \quad (43)$$

Rejeitar a hipótese nula conduz à conclusão de que, estatisticamente, o desempenho dos filtros é diferente para um nível de significância de 5% (valor do nível de significância adotado para a realização dos testes de hipótese).

Outro parâmetro importante é o valor de prova, p . O valor de prova é o menor valor que α pode assumir para que a hipótese H_0 continue a ser rejeitada. No caso em que o valor de prova é superior a significância do teste, não se pode rejeitar a hipótese nula. O valor de prova é calculado em cada teste e quanto mais baixo for o seu valor, mais certeza se terá na rejeição da hipótese nula.

Tendo em consideração problemas de inicialização dos filtros discutidos na fase de otimização, optou-se, na fase de comparação, por excluir os primeiros 200 pontos do cálculo do erro. Esta abordagem permite que os filtros que sofram de erros significativos durante a inicialização tenham tempo para melhorar as suas estimativas, conduzindo a uma análise mais justa. Esta análise é também mais realista porque na prática, os filtros irão ser utilizados por longos períodos de tempo e os erros de inicialização serão negligenciáveis.

5.2.2 -Resultados obtidos no teste dos vários filtros estudados

Apresentam-se nesta secção os resultados decorrentes do teste dos filtros estudados. Como já foi referido, inicialmente a comparação será feita entre 7 filtros baseados em dados. São eles: a janela móvel *online*, o filtro exponencial, o filtro de Holt-Winters, o filtro de Butterworth, o filtro de Chebyshev, o filtro elítico e por fim, o filtro multiescala *online*. Os sinais utilizados serão o *HeaviSine*, o *Doppler*, os *Blocks*, os *Bumps* e a resposta da junção de dois sistemas de primeira ordem. Cada sinal será contaminado com 4 níveis de ruído, correspondentes a um nível baixo de ruído ($SNR=25$), um nível médio ($SNR=15$), um médio-alto ($SNR=10$) e um nível alto ($SNR=5$).

A título ilustrativo, apresentam-se na Tabela 3 e na Tabela 4 os **Erro! A origem da referência não foi encontrada**. resultados do teste correspondentes a diferentes combinações sinais/ruído. A Figura 8 apresenta o sinal real, o medido e o filtrado para o caso do sinal *Doppler*,

Resultados

para o sinal *Blocks*, para a resposta do sistema de primeira ordem e para os *Bumps*. A filtragem é realizada utilizando o filtro exponencial, o filtro multiescala *online*, o filtro de Holt e o filtro de Butterworth, respetivamente. Tabelas completas com os restantes resultados do teste podem ser consultadas no Anexo II.

Tabela 3 - Resultados do teste dos filtros sem modelos utilizando o sinal de HeaviSine com SNR=5 e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.0089 <1E-5	-0.009 <1E-12	-0.0322 <1E-10	-0.0426 <1E-12	-0.0647 <1E-14	-0.1559 <1E-19
EWMA	-0.0089 <1E-5	-	-0.0179 <1E-8	-0.0411 <1E-11	-0.0515 <1E-13	-0.0736 <1E-16	-0.1648 <1E-19
BWT	0.009 <1E-12	0.0179 <1E-8	-	-0.0231 <1E-7	-0.0335 <1E-9	-0.0557 <1E-12	-0.1468 <1E-18
CBS	0.0322 <1E-10	0.0411 <1E-11	0.0231 <1E-7	-	-0.0104 <1E-12	-0.0325 <1E-7	-0.1237 <1E-17
ELI	0.0426 <1E-12	0.0515 <1E-13	0.0335 <1E-9	0.0104 <1E-12	-	-0.0221 <1E-5	-0.1133 <1E-17
OLMS	-0.0647 <1E-14	0.0736 <1E-16	0.0557 <1E-12	0.0325 <1E-7	0.0221 <1E-5	-	-0.0912 <1E-14
HF	0.1559 <1E-19	0.1648 <1E-19	0.1468 <1E-18	0.1237 <1E-17	0.1133 <1E-17	0.0912 <1E-14	-

Nota: em cada célula da tabela, a primeira entrada corresponde a média das diferenças (\bar{d}) entre o filtro que se encontra na linha e o filtro da coluna. O número imediatamente abaixo corresponde ao valor de prova (p).

Tabela 4 - Resultados do teste dos filtros sem modelos utilizando o sinal de blocos com SNR=15 e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.0158 <1E-21	-0.0265 <1E-22	-0.0603 <1E-24	-0.0531 <1E-23	0.0793 <1E-25	-0.0169 <1E-16
EWMA	-0.0158 <1E-21	-	-0.0423 <1E-24	-0.0761 <1E-25	-0.0689 <1E-25	0.0635 <1E-25	-0.0327 <1E-23
BWT	0.0265 <1E-22	0.0423 <1E-24	-	-0.0338 <1E-21	-0.0266 <1E-20	0.1058 <1E-25	0.0097 <1E-12
CBS	0.0603 <1E-24	0.0761 <1E-25	0.0338 <1E-21	-	0.0072 <1E-21	0.1396 <1E-27	0.0434 <1E-20
ELI	0.0531 <1E-23	0.0689 <1E-25	0.0266 <1E-20	-0.0072 <1E-21	-	0.1324 <1E-27	0.0362 <1E-19
OLMS	-0.0793 <1E-25	-0.0635 <1E-25	-0.1058 <1E-25	-0.1396 <1E-27	-0.1324 <1E-27	-	-0.0961 <1E-25
HF	0.0169 <1E-16	0.0327 <1E-23	-0.0097 <1E-12	-0.0434 <1E-20	-0.0362 <1E-19	0.0961 <1E-25	-

Nota: em cada célula da tabela, o primeiro número corresponde a média das diferenças (\bar{d}) entre o filtro que se encontra na linha e o filtro da coluna. O número imediatamente abaixo corresponde ao valor de prova (p).

Resultados

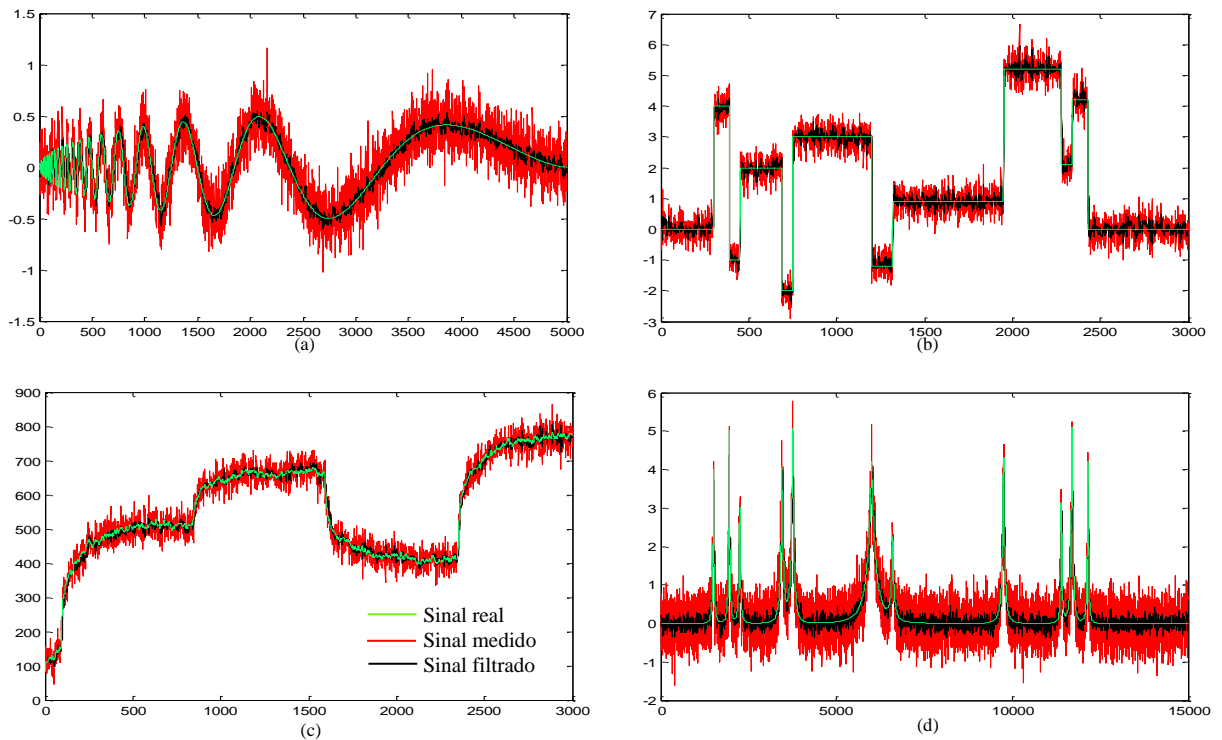


Figura 8 - Sinais sem ruído, medidos e filtrados. (a) corresponde ao sinal de Doppler com SNR=5 filtrado com o filtro exponencial; (b) corresponde aos blocos, com SNR=15, filtrado com o filtro multiescala *online*; (c) é a resposta da junção dos sistemas de primeira ordem com SNR=15, filtrado com o filtro de Holt; (d) corresponde aos *Bumps*, com SNR=5 e filtrado com o filtro de Butterworth.

A análise da Tabela 3 e da Tabela 4 permite facilmente concluir qual o filtro com melhor desempenho. Em cada célula destas tabelas, a primeira entrada corresponde à média das diferenças (\bar{d}) entre a metodologia de filtragem que se encontra na linha e a metodologia que se encontra na coluna. Assim sendo, o melhor filtro é aquele cuja coluna/linha é composta apenas por valores positivos/negativos, i.e., os erros produzidos pelos filtros que se encontra nas linhas/colunas são superiores. O número imediatamente abaixo da média das diferenças é o valor de prova (p), que deverá ser inferior a 5% para que se possa afirmar que existe uma diferença nos desempenhos, caso contrário, as metodologias são consideradas equivalentes.

Para resumir o elevado volume de resultados provenientes da aplicação do procedimento de teste acima descrito a todos os tipos de sinal e níveis de ruído, contabilizou-se o número de vezes que um dado método obteve um desempenho superior aos seus oponentes (“vitórias”), bem como aquelas em que este foi inferior (“derrotas”) e estatisticamente equivalente, i.e., quando o correspondente valor de prova for superior a 0.05 (“empates”). A Figura 9 e a Figura 10 sumarizam os resultados obtidos desta análise, para os filtros baseados em dados.

Resultados

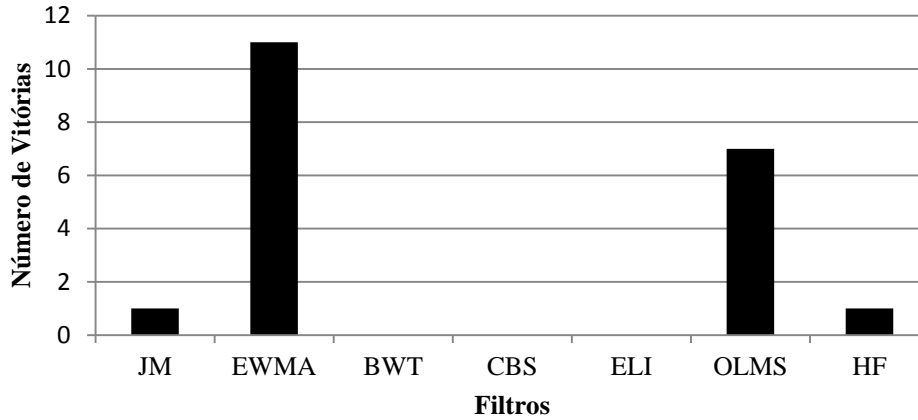


Figura 9 - Número de vezes que cada filtro sem modelo apresentou o menor erro (número de vitórias).

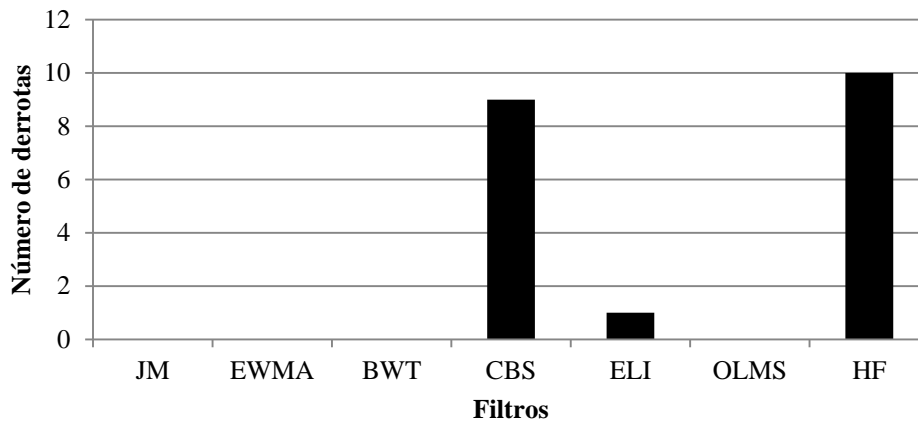


Figura 10 - Número de vezes que cada filtro sem modelo apresentou o maior erro (número de derrotas).

Como se pode observar na Figura 9 o filtro exponencial é aquele que mais vezes apresentou um melhor desempenho nos casos estudados, seguido pelo filtro multiescala *online*. No extremo oposto encontra-se o filtro de Holt com um maior número de derrotas (Figura 10), seguido do filtro de Chebyshev. Portanto, destaca-se desta pré-seleção o filtro exponencial e o multiescala *online*, que irão ser utilizados no estudo comparativo envolvendo metodologias mais complexas, que recorrem a modelos na sua implementação.

A comparação com metodologias mais complexas inicia-se com a inclusão do filtro de Kalman. Uma das mais-valias referidas é a sua capacidade de incorporar informação sobre os fenómenos em causa na forma de um modelo matemático, e logo as variáveis de entradas do sistema. Em determinadas condições (sistemas lineares, invariantes no tempo e ruído branco Gaussiano) este filtro conduz à solução ótima para o problema de estimação do valor atual, i.e., à filtragem ótima de dados. Consideraram-se dois tipos de sinais: a entrada e a saída da junção de dois sistemas de primeira ordem anteriormente mencionados e outro modelo que resulta da modelação de um reator contínuo perfeitamente agitado (Shi *et al* 2000) e que apresenta *overshoot* e resposta inversa. Este sistema é descrito pelas seguintes matrizes (ver o seu significado nas equações: (20) e (21)):

Resultados

$$A = \begin{bmatrix} 0.68654 & -5.046 \times 10^{-4} \\ 64.182 & 1.0177 \end{bmatrix}, B = \begin{bmatrix} 1.726 \times 10^{-4} & -1.0647 \times 10^{-5} \\ -3.5311 \times 10^{-2} & 4.0478 \times 10^{-2} \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Utilizar-se-á a primeira saída deste sistema para efeitos de filtragem. Os sinais anteriormente descritos (*HeaviSine*, *Doppler*, *Blocks*, etc.) não foram utilizados porque esses não foram gerados por um modelo dinâmico. Apresenta-se na Figura 11 o número de vitórias e na Figura 12 o número de derrotas obtidas pelo filtro exponencial, o de Kalman e o multiescala *online* quando utilizados na filtragem de dados obtidos da simulação dos dois sistemas acima descritos. Uma vez mais utilizam-se 4 níveis de ruído: $SNR=5, 10, 15$ e 25 .

Na Figura 11 observa-se que o desempenho do filtro de Kalman é bastante bom, tendo sido o filtro com menor erro em cerca de 75% dos casos. Apesar de ter obtido algumas derrotas, fica demonstrada a sua excelente performance e a sua aplicabilidade a sistemas descritos por um modelo de espaço de estados. A Figura 13 apresenta o sinal sem ruído, o medido e o filtrado para o sistema com *overshoot*, utilizando o filtro de Kalman e o filtro exponencial.

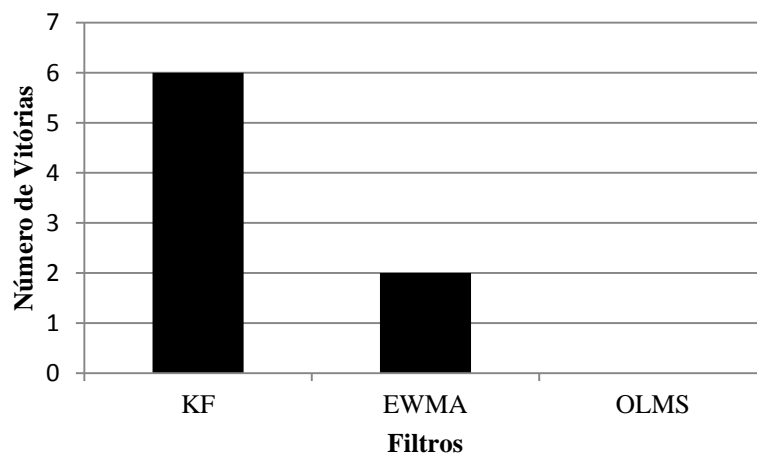


Figura 11 - Número de vitórias obtidas na filtragem da junção de dois sistemas de primeira ordem e na filtragem do sistema com *overshoot*.

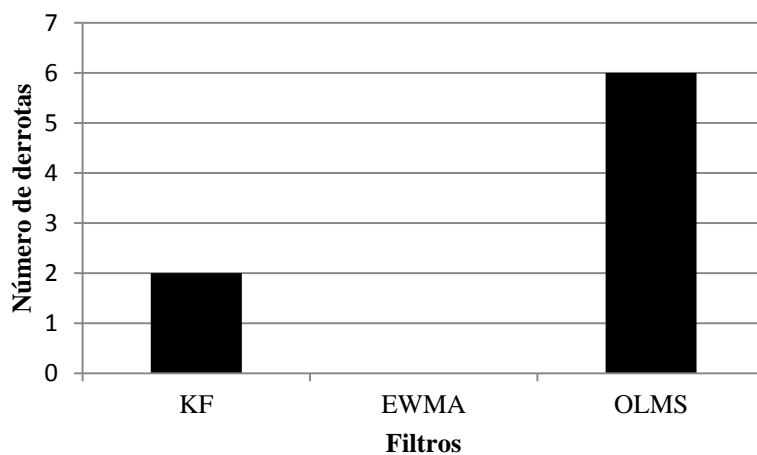


Figura 12 - Número de derrotas obtidas na filtragem da junção de dois sistemas de primeira ordem e na filtragem do sistema com *overshoot*.

Resultados

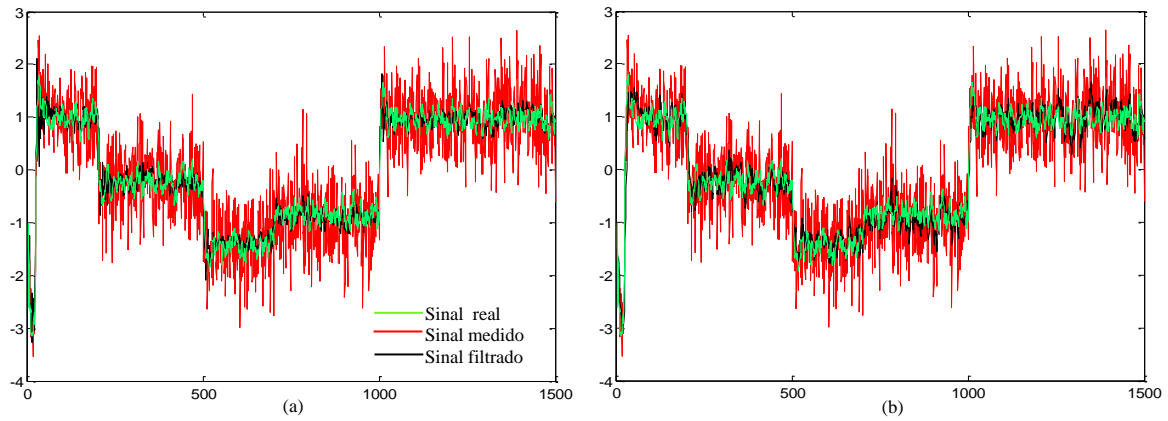


Figura 13 - Sinais sem ruído, medidos e filtrados decorrentes da simulação de um sistema com *overshoot*, com SNR=5 e 10 iterações. O sinal filtrado em (a) é obtido pela aplicação do filtro de Kalman e em (b) pela aplicação do filtro exponencial.

Neste trabalho considerou-se ainda a análise e teste do filtro de Kalman multiescala. Recorreremos aos sinais estudados durante a pré-seleção, e nesses casos utilizaremos apenas os filtros exponencial e multiescala *online*. A Figura 14 apresenta o número de vitórias e a Figura 15 apresenta o respectivo número de derrotas.

Como se pode observar na Figura 14 o desempenho do filtro de Kalman multiescala não é o esperado. Uma vez que a teoria e a construção dos filtros são mais complexas, esperar-se-ia que os resultados fossem melhores. No entanto, tal não acontece e algumas justificações serão apresentadas no capítulo da discussão.

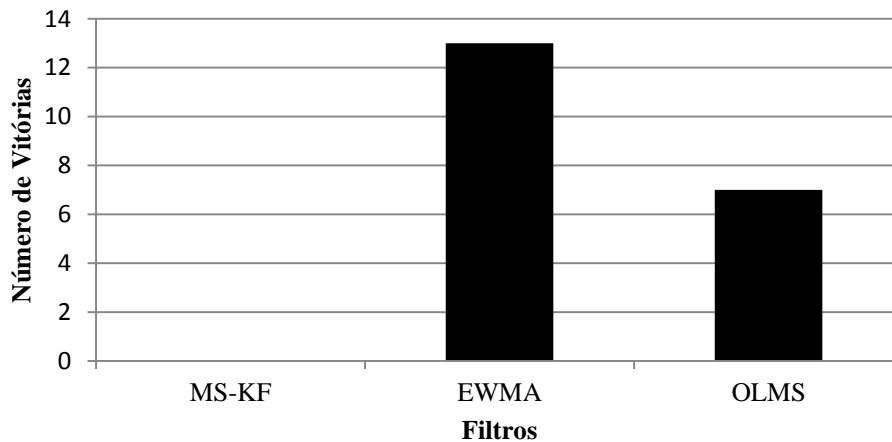


Figura 14 - Número de vitórias obtidas na filtragem dos sinais sem modelo (*HeaviSine*, *Doppler*, *Bumps* e *Blocks*) e na filtragem do sinal obtido pela junção de dois sistemas de primeira ordem.

Resultados

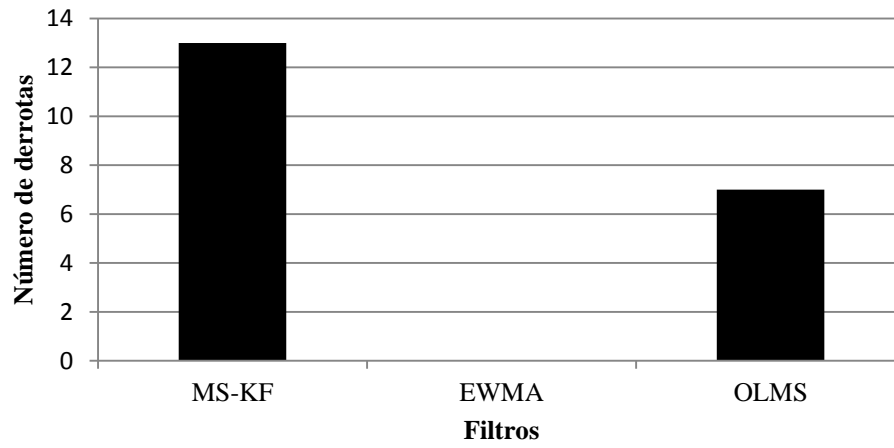


Figura 15 - Número de derrotas obtidas na filtragem dos sinais sem modelo (*HeaviSine*, *Doppler*, *Bumps* e *Blocks*) e na filtragem do sinal obtido pela junção de dois sistemas de primeira ordem.

6 - Discussão dos resultados

Neste capítulo é feita a discussão dos resultados apresentados anteriormente, justificando-os e procurando obter linhas de orientação que auxiliem na escolha dos filtros digitais. A discussão inicia-se com os filtros baseados em dados. De seguida, aborda-se o filtro de Kalman e por fim, o filtro de Kalman multiescala.

6.1 -Discussão dos métodos de filtragem baseados em dados

Os filtros baseados em dados são de implementação mais simples e são utilizados em muitas aplicações. A Figura 9 permite destacar o filtro exponencial e o filtro multiescala *online* como sendo aqueles que minimizam a diferença entre o sinal real e o sinal filtrado. A atribuição de um peso à medição atual e a medições passadas e a capacidade de aceder ao conteúdo multiescala de um sinal permitiu que o filtro exponencial e o filtro multiescala *online* conseguissem extrair grande parte do ruído e mesmo assim responder a alterações rápidas no sinal real. Estas abordagens foram mais eficazes do que aquelas que se centraram no domínio da frequência ou que atribuem o mesmo peso às medições passadas e às mais recentes. O filtro de Holt obteve o pior desempenho e portanto não deve ser utilizado na remoção de ruído branco, contudo, em todos os casos estudados, conduziu ao menor erro de previsão. Este facto justifica-se pela natureza do filtro, que foi construído centrando-se no problema da previsão. Conclui-se assim que o erro de previsão é adequado para a sintonização dos filtros mas não deve ser utilizado na comparação dos desempenhos, principalmente quando se analisam métodos especializados em previsão.

Para evidenciar algumas das diferenças entre o filtro exponencial e o multiescala *online*, apresenta-se novamente o número de vitórias de cada um, agrupadas em dois grupos: um com uma menor contribuição do ruído ($SNR=15$ e 25) e outro com uma maior contribuição ($SNR=5$ e 10). A Figura 16 apresenta o número de vitórias para $SNR=5$ e 10 e a Figura 17 apresenta o número de vitórias para $SNR=15$ e 25 .

Como se observa na Figura 16 e na Figura 17, o filtro exponencial apresenta um maior número de vitórias para valores de SNR baixos, enquanto que o filtro multiescala *online* aparenta ser mais adequado para $SNRs$ elevados. Esta complementaridade permite que os filtros constituam um conjunto adequado para a comparação com metodologias mais complexas.

É importante ainda salientar o benefício da análise multiescala, uma vez que os sinais apresentados não possuem características multiescala muito marcantes. A junção de alguns degraus, dos *Bumps* e do sinal de *HeaviSine* produz um sinal com um espectro de frequências alargado e que possui características temporais significativas. A Figura 18 apresenta este sinal filtrado com o filtro exponencial e com o filtro multiescala *online*, para um $SNR=25$.

Discussão dos resultados

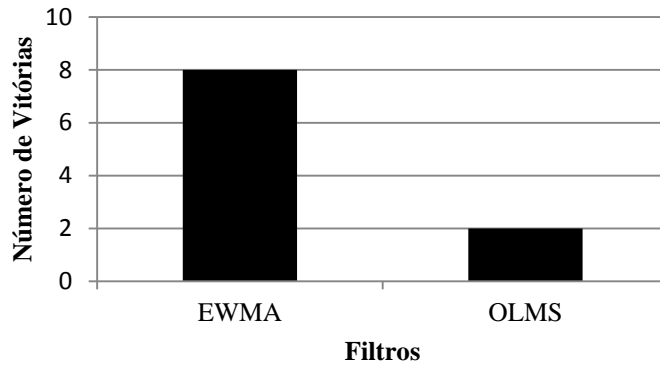


Figura 16 - Número de vitórias dos filtros exponencial e multiescala *online* para SNR=5 e 10.

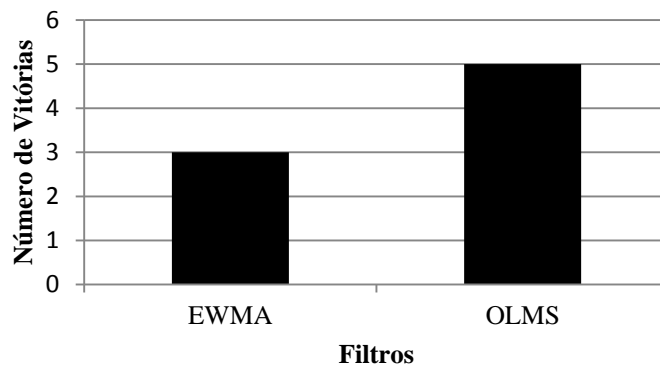


Figura 17 - Número de vitórias dos filtros exponencial e multiescala *online* para SNR=15 e 25.

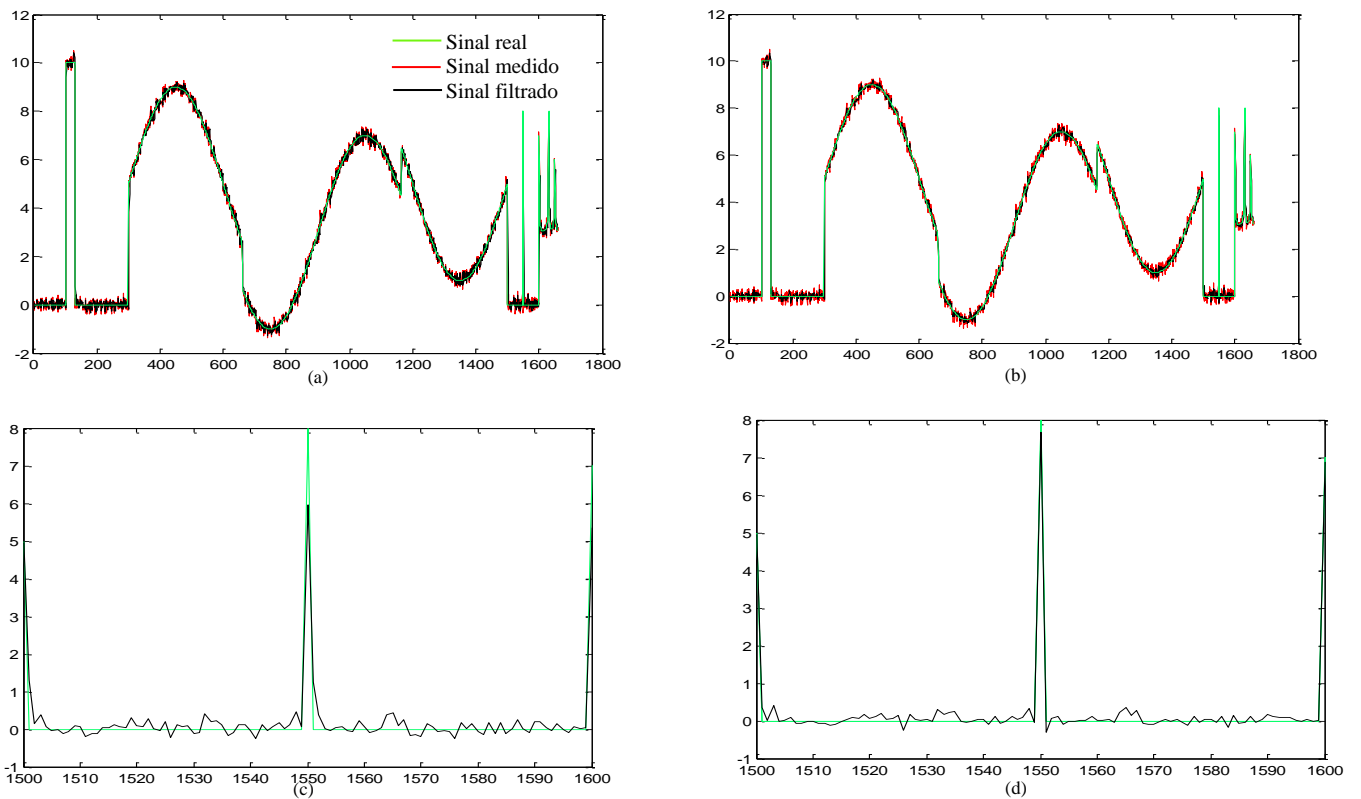


Figura 18 - Sinal com características multiescala filtrado com o filtro exponencial (a) e com o filtro multiescala *online* (b). O intervalo apresentado em (c) e (d) destaca as diferenças entre os filtros.

A Figura 18 permite evidenciar os compromissos associados ao filtro exponencial que só tem em consideração uma única escala. Como se pode observar nas tabelas de otimização (Anexo I), sinais como o de *HeaviSine* são melhor filtrados com valores baixos de α , enquanto que sinais como os *Bumps* tendem a originar valores de α mais elevados. Assim sendo, o valor de α obtido na otimização não é ótimo para nenhum dos casos, sendo que este se adapta ao valor intermédio que minimize o erro. É notório na Figura 18 que α assume um valor elevado, com o intuito de acompanhar as mudanças repentinas no sinal, contudo a sua capacidade para remover ruído é degradada (o sinal filtrado (a preto) sobrepõe-se muitas vezes com o sinal medido (a vermelho)). Quanto ao filtro multiescala *online*, observa-se que a quantidade de ruído removida é maior, sendo particularmente interessante observar a amostra 1550. O filtro multiescala *online* acompanha muito mais eficazmente este pico, ao contrário do filtro exponencial que o atenua em demasia.

Por fim, para ter maior sensibilidade relativamente aos restantes filtros baseados em dados, apresenta-se na Figura 19 o respetivo número de vitórias. Estes resultados são calculados tendo em conta os filtros que não se destacaram, nem pela positiva nem pela negativa, nomeadamente: o filtro da média da janela móvel, o filtro de Butterworth e o filtro elíptico. Observa-se que o filtro da média da janela móvel é o melhor dos três. Pode-se dizer que em geral os filtros que analisam exclusivamente a escala de frequências (Butterworth, Chebyshev e elíptico) não conduziram a bons resultados, sendo que a justificação está na natureza do ruído branco, que possui a mesma energia distribuída por todas as frequências, e não limitada a uma certa banda de frequências. A definição de uma frequência de corte acima da qual todas as frequências são eliminadas elimina o ruído com frequências mais elevada. No entanto, o ruído cuja frequência é inferior a essa frequência de corte permanece, afetando as frequências que constituem o sinal real.

A análise dos filtros baseados em dados permitiu destacar alguns pontos importantes. Na remoção de ruído branco, o filtro exponencial e o multiescala *online* são preferíveis. O filtro exponencial produziu o menor erro num maior número de ocasiões, sendo que a sua performance é superior para valores de SNR baixos (5 e 10). O filtro multiescala *online* é mais adequado a valores de SNR elevados (15 e 25), sendo que o seu desempenho é significativamente maior para sinais com uma forte componente multiescala. A utilização do filtro de Holt, do filtro de Butterworth, do de Chebyshev e do elíptico para a remoção de ruído branco deve ser evitada.

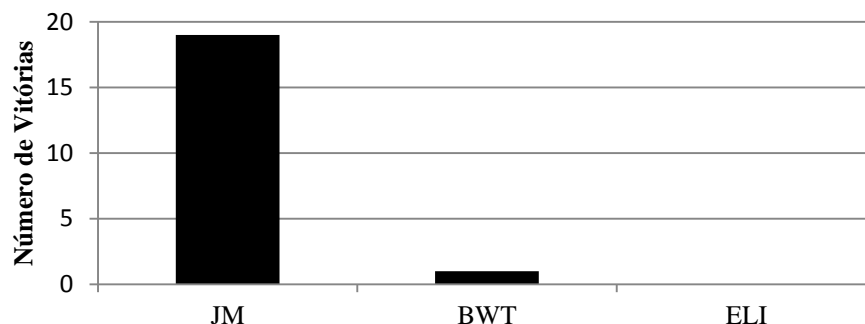


Figura 19 - Número de vitórias tendo em conta apenas o filtro da média da janela móvel, o de Butterworth e o elíptico.

6.2 -Discussão dos resultados obtidos da comparação entre o filtro exponencial, o filtro multiescala *online* e o filtro de Kalman

A aplicação do filtro de Kalman conduziu aos resultados esperados. Na Figura 11 observa-se que o filtro apresentou a menor diferença entre o valor filtrado e o valor real, sendo que o filtro exponencial apresenta a segunda melhor performance. Vale a pena salientar, uma vez mais, a capacidade que o filtro de Kalman possui de combinar a informação à entrada do sistema. Caso tal informação não estivesse disponível, as estimativas produzidas pelo filtro não seriam as melhores, o que se comprova na Figura 20.

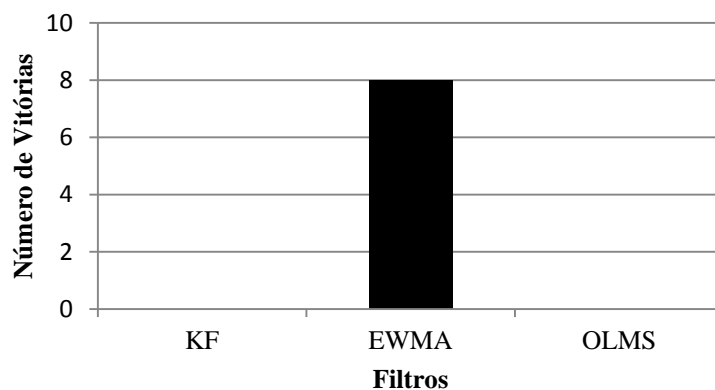


Figura 20 -Número de vitórias obtidas pelo filtro exponencial, pelo de Kalman e pelo multiescala *online*. A filtragem utilizando o filtro de Kalman não teve em conta as entradas ao sistema.

Conclui-se assim que nos casos em que o sistema é bem descrito por um modelo de espaço de estados e existem sinais de entrada e saída disponíveis, deve-se utilizar o filtro de Kalman. Caso contrário, deve-se optar por metodologias como o filtro exponencial ou o filtro multiescala *online*, consoante o nível de ruído e a natureza do sinal (com características a uma só escala ou multiescala). A possibilidade de utilizar outros métodos que não o N4SID é também uma alternativa por explorar e que poderia conduzir a resultados ainda melhores, caso esses métodos estimassem com maior exatidão o sistema em causa. Como se pode observar pelos resultados da otimização (Anexo I), a ordem estimada para o sistema é sempre muito superior a sua ordem real, o que pode ser uma indicação de que fenómenos estocásticos poderão estar a ser modelados, prejudicando o desempenho.

6.3 - Discussão dos resultados obtidos com o filtro exponencial, o de Kalman multiescala e o multiescala *online*

O filtro de Kalman multiescala não apresentou os resultados esperados, como se pode observar na Figura 14. Em nenhuma ocasião o filtro apresentou o melhor desempenho, muito pelo contrário, foi o que mais vezes conduziu ao maior erro entre o sinal filtrado e o real. Algumas causas poderão estar na origem deste fraco desempenho. A primeira prende-se essencialmente com a natureza monoescala dos sistemas simulados que não penaliza a ausência de uma estrutura que descreva as diversas escalas analisadas. Por outras palavras, os sistemas e os sinais estudados estão codificados essencialmente numa só escala e por isso a análise multiescala não traz muitas vantagens. Outra justificação advém da utilização do soft *thresholding* numa janela de dimensão fixa, que quando é reduzida, poderia conduzir a uma excessiva eliminação de coeficientes, i.e., ao contrário do que acontece no filtro multiescala *online* em que a janela diádica é a maior possível, no caso do filtro de Kalman multiescala esta tem uma dimensão fixa e se for demasiadamente pequena poderá conduzir a eliminação de coeficientes que transportam informação relevante sobre o sistema.

Com o intuito de contornar as fracas características multiescala dos sinais estudados, recorreu-se a um novo sinal com características multiescala mais marcantes. Este sinal é obtido a partir da simulação de um reator não isotérmico (Nounou (2006)), e o autor que o apresenta conclui que a utilização do filtro de Kalman multiescala é mais adequada que a utilização do filtro de Kalman clássico. Assim sendo, este sistema foi simulado e os resultados da comparação entre o filtro de Kalman multiescala, o filtro exponencial e o filtro multiescala *online* são apresentados na Figura 21 e na Figura 22. O sistema simulado é descrito pelas seguintes matrizes (ver o seu significado nas equações: (20) e (21)).

$$A = \begin{bmatrix} 0.65 & 0.1 \\ 0.075 & 0.8048 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0.2835 \end{bmatrix}, C = [1 \quad 1]$$

Como se pode observar na Figura 21 e na Figura 22, o filtro de Kalman multiescala, uma vez mais, não apresentou o melhor desempenho. Tendo em conta estes resultados, a causa mais provável para o fraco desempenho deste filtro reside na utilização do método N4SID. Nounou (2006) concluiu que o filtro de Kalman multiescala é o mais adequado, contudo, a aplicação deste filtro foi feita com o total conhecimento de todas as matrizes que caracterizam o sistema, i.e., as matrizes não foram estimadas como acontece na metodologia aqui adotada. Assim sendo, apesar da metodologia de Kalman multiescala poder ser vantajosa, a sua eficácia não ficou demonstrada para casos em que as matrizes do sistema têm de ser estimadas. A utilização de melhores métodos de identificação de sistemas é então uma alternativa a explorar.

A análise do filtro de Kalman multiescala permitiu concluir que este não deve ser utilizado para a remoção de ruído branco. Alternativas ao N4SID deverão ser exploradas com o objetivo de melhorar o seu desempenho. Assim sendo, as metodologias anteriormente mencionadas como o filtro de Kalman, o filtro exponencial e o filtro multiescala *online* deverão ser a primeira

Discussão dos resultados

escolha para a filtragem de dados contaminados com ruído branco. A escolha da metodologia mais adequada não é de todo trivial, mas pelo estudo efetuado, algumas linhas orientadoras se destacam. A primeira é a de que a otimização dos filtros pode ser feita pela minimização do erro de previsão, contudo, este critério não é adequado para a comparação de diferentes metodologias de filtragem. O filtro de Kalman deve ser a primeira escolha, desde que as entradas do sistema estejam disponíveis. No caso de se pretender utilizar metodologias mais simples, o filtro multiescala *online* e o filtro exponencial são os mais aconselháveis, sendo o último o mais simples deste lote e de implementação trivial. A escolha entre eles pode basear-se na quantidade de ruído que afeta o sinal ou então por analogia com os resultados aqui expostos (Anexo II). Comparando os sinais produzidos pelo sistema em análise com aqueles aqui estudados, o gestor de processos pode escolher um filtro que terá um desempenho muito próximo do ótimo. Naturalmente, sinais com características multiescala deverão ser filtrados com o filtro multiescala *online* que demonstrou uma melhor aptidão para estes tipos de sinais.

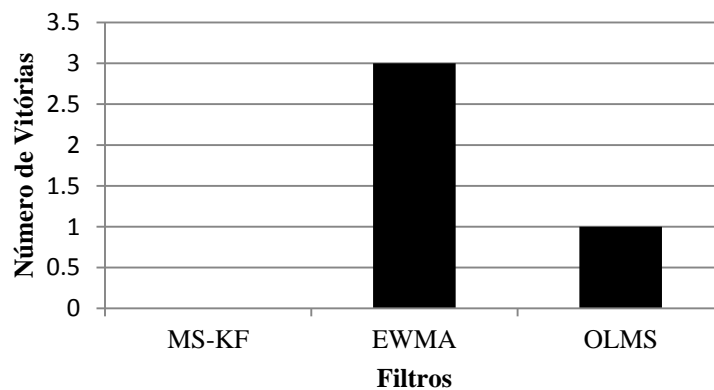


Figura 21 - Número de vitórias obtidas na filtragem do reator não isotérmico.

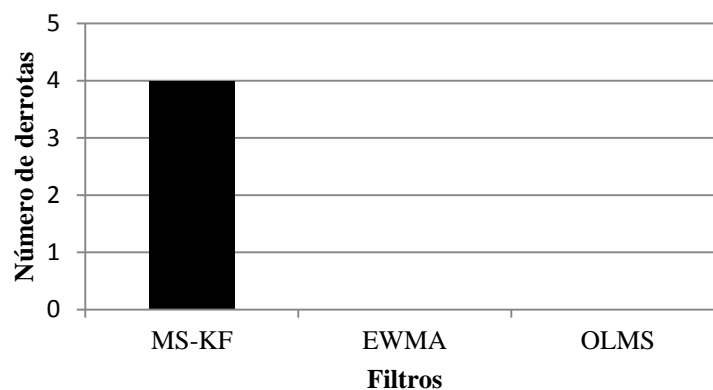


Figura 22 - Número de derrotas obtidas na filtragem do reator não isotérmico.

7 - Conclusões e perspectivas futuras

A filtragem de dados é uma tarefa essencial em qualquer processo de aquisição de dados, permitindo remover o ruído que advém das mais variadas fontes (instrumentos de medição, erros humanos, etc.). A presente dissertação teve por objetivo avaliar o desempenho de diferentes metodologias de filtragem na remoção de ruído branco, utilizando alguns sinais clássicos da literatura e outros provenientes da simulação de sistemas. As simulações foram efetuadas com 4 níveis de ruído ($SNR=5$, 10, 15 e 25).

Os filtros digitais dividem-se em dois grupos: os filtros baseados em dados e os filtros que, para além dos dados, incorporam um modelo do sistema. O estudo comparativo iniciou-se com a pré-seleção de 7 filtros baseados em dados: a média da janela móvel, o filtro exponencial, o filtro Butterworth, o filtro Chebyshev, o filtro elíptico, o filtro multiescala *online* e o filtro de Holt. A implementação de qualquer uma destas metodologias está condicionada pela especificação de parâmetros próprios de cada filtro. Assim sendo, numa primeira fase os filtros foram otimizados para que a filtragem conduzisse a estimativas onde a presença de ruído fosse minimizada, escolhendo como o melhor parâmetro aquele que minimiza o erro de previsão. O baixo valor do desvio percentual entre o parâmetro obtido durante a otimização e o seu valor ideal permitiu concluir que esta metodologia pode e deve ser utilizada na otimização de filtros. De seguida foi feita a comparação, recorrendo a testes de hipóteses que permitiram diferenciar, estatisticamente, o desempenho dos filtros. Desta pré-seleção destacaram-se, pela positiva, o filtro exponencial e o filtro multiescala *online*, enquanto que se concluiu que a utilização do filtro de Holt ou do filtro de Chebyshev não é eficaz na remoção de ruído branco. Contudo o filtro de Holt apresentou sempre o menor erro de previsão em todos os casos estudados, o que permite concluir que a utilização do erro de previsão para a comparação de metodologias de filtragem não é adequada. Para os sinais estudados, o desempenho do filtro exponencial foi melhor para situações onde o nível de ruído era elevado ($SNR=5$ e 10) e o filtro multiescala *online* foi mais adequado para níveis baixos de ruído ($SNR=15$ e 25). Este último ainda demonstrou claras vantagens quando o sinal continha características multiescala, mantendo uma resposta rápida quando ocorreram alterações bruscas no sinal e ao mesmo tempo removendo uma grande quantidade de ruído.

Após a pré-seleção dos filtros baseados em dados, considerou-se as metodologias de filtragem que também têm em conta um modelo do processo. O filtro de Kalman destaca-se pela estimativa ótima que produz, desde que certas condições sejam satisfeitas (ruído branco, modelo de espaço de estados, etc.), sendo que na sua implementação recorreu-se a um método de identificação de sistemas (N4SID). Estudaram-se dois modelos de espaço de estados: um que resulta da junção de dois sistemas de primeira ordem e outro sistema com *overshoot* e resposta inversa. Sem nenhuma surpresa, esta metodologia forneceu estimativas mais próximas do valor integral da variável medida em 75% dos casos estudados e portanto a sua aplicação é altamente recomendável, desde que as condições o permitam.

Conclusões e perspectivas futuras

A utilização do filtro de Kalman multiescala não conduziu aos resultados esperados. Apesar da sua capacidade de estimar os coeficientes a cada escala de forma ótima, o erro apresentado foi sempre superior ao do filtro exponencial e ao do filtro multiescala *online*. Para os sinais estudados, a utilização deste filtro não é recomendada, sendo que outras metodologias de identificação de sistemas deverão ser utilizadas para tentar melhorar o seu desempenho. Este fraco desempenho é também justificado pelas fracas características multiescala de alguns dos sinais estudados

Como perspectivas futuras destaca-se o potencial de combinação das diferentes metodologias de filtragem. Assim como a combinação do filtro de Kalman e do filtro multiescala *online* conduziu ao filtro de Kalman multiescala, outras combinações poderão ser efetuadas, originando metodologias de filtragem mais eficazes. Seria também interessante comparar o desempenho das metodologias aqui expostas, aquando do tratamento de sinais contaminados com um ruído que oscila numa dada frequência. Neste caso, talvez filtros como o Butterworth, Chebyshev e elíptico se pudessem destacar pela positiva. O tratamento de sistemas não lineares e a robustez à *outliers* são também áreas de elevado interesse. Por fim, a utilização de outros métodos de identificação de sistemas poderá melhorar o desempenho do filtro de Kalman multiescala.

Bibliografia

- Akesson, B. M.; Jorgensen, J. B.; Poulsen, N. K.; Jorgensen, S. B. (2008) - "*A generalized autocovariance least-squares method for Kalman filter tuning*" - Journal of Process Control, 18, pp. 769-779.
- Bakshi, B. R. (1999) - "*Multiscale Analysis and Modeling using Wavelets*" - Journal of Chemometrics, 13, pp. 415-434.
- Brown, R. G., Hwang, P. Y. (1997) - "*Introduction to Random Signals and Applied Kalman Filtering*" - 2° ed.. New York: John Wiley & Sons.
- Chatfield, C., Yar, M. (1988) - "*Holt-Winters Forecasting: Some Practical Issues*" - Journal of the Royal Statistical Society, 37(2), pp. 129-140.
- Chui, C. K., Chen, G. (2009) - "*Kalman Filtering with Real-Time Applications*" - 4 ed.. Verlag Berlin: Springer.
- Daubechies, I. (1988) - "*Orthonormal Bases of Compactly Supported Wavelets*" - Communications on Pure and Applied Mathematics, 41, pp. 909-996.
- Doymaz, F.; Bakhtazad, A.; Romagnoli, J. A.; Palazoglu, A. (2001) - "*Wavelet-based robust filtering of process data*" - Computers and Chemical Engineering, 25, pp. 1549-1559.
- Favoreel, W.; Moor, B.; Overschee, P. V. (2000) - "*Subspace state system identification for industrial processes*" - Journal of Process Control, 10, pp. 149-155.
- Gelper, S.; Fried, R.; Croux, C. (2010) - "*Robust Forecasting with Exponential and Holt-Winters Smoothing*" - 29, pp. 285-300.
- Graps, A. (1995) - "*An introduction to Wavelets*" - 2(2), pp. 1-18.
- Haseltine, E. L., Rawlings, J. B. (2005) - "*Critical Evaluation of Extended Kalman Filtering and Moving-Horizon Estimation*" - Ind. Eng. Chem. Res., 44, pp. 2451-2460.
- Kalman, R. E. (1960) - "*A New Approach to Linear Filtering and Prediction Problems*" - Journal of Basic Engineering, 82, pp. 35-45.
- Karu, Z. Z. (2001) - "*Signals and Systems made Ridiculously Simple*" - Cambridge, MA: ZiZi Press.
- Kohler, T., Lorenz, D. (2005) - "*A comparison of denoising methods for one dimensional time series*" - pp. 1-15.
- Krakovsky, V. Y. (2006) - "*Moving-Window Discrete Fourier Transform*" - Journal of Real-Time Image Processing, 1, pp. 153-161.

Bibliografia

- Ljung, L. (1987) - "*System Identification: Theory for the User*" - Englewood Cliffs, New Jersey: PTR Prentice Hall.
- Lyons, R. (2001) - "*Understanding Digital Signal Processing*" - Upper Saddle River, New Jersey: Prentice Hall PTR.
- Mallat, S. G. (1989) - "*A Theory for Multiresolution Signal Decomposition: The Wavelet Representation*" - Transactions on Pattern analysis and machine intelligence, 2(7), pp. 674-693.
- Montgomery, D. C., Runger, G. C. (2003) - "*Applied Statistics and Probability for Engineers*" - New York: John Wiley & Sons.
- Nounou, M. N. (2006) - "*Enhanced State Estimation using multiscale Kalman filter*" - Proceedings of the 45th IEEE Conference on Decision & Control, pp. 1679-1684.
- Nounou, M. N., Bakshi, B. R. (1999) - "*On-Line Multiscale Filtering of Random and Gross Errors without Process Models*" - 45(5), pp. 1041-1058.
- Oppenheim, A. V.; Willsky, A. S.; Nawab, S. H. (1997) - "*Signals and Systems*" - Upper Saddle River, New Jersey: Prentice Hall.
- Overschee, V. P., Moor, B. (1994) - "*N4SID: Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems*" - Automatica, 30(1), pp. 75-93.
- Padonou, E.; Roustant, O.; Lutz, M. (2013) - "*Robust monitoring of an industrial IT system in presence of structural change*" - pp. 1-23.
- Qin, S. J. (2006) - "*An overview of subspace identification*" - Computers and Chemical Engineering, 30, pp. 1502-1513.
- Reis, M. S. (2009) - "*A multiscale empirical modeling framework for system identification*" - Journal of Process Control, 19, pp. 1546-1557.
- Reis, M. S.; Saraiva, P. M.; Bakshi, B. R. (2009) - "*Denoising and Signal-to-Noise Ratio Enhancement: Wavelet transform and Fourier transform*" - Vol. 2. Oxford: Elsevier.
- Sardy, S.; Tseng P.; Bruce, A. (2001) - "*Robust Wavelet Denoising*" - Transactions on Signal Processing, 49(6), pp. 1146-1152.
- Shi, R., MacGregor, J. F. (2000) - "*Modeling of dynamic systems using latent variable and subspace methods*" - Journal of Chemometrics, 14, pp. 423-439.
- Smith, S. W. (1997) - "*The scientist and engineer's guide to digital signal processing*" - San Diego: California Technical Publishing.
- Welch, G., & Bishop, G. (2006) - "*An Introduction to th Kalman Filter*" -. pp. 1-16.

Anexo I

Resultados da otimização dos parâmetros dos filtros

Apresenta-se neste anexo, as tabelas que contém os resultados decorrentes da otimização dos filtros.

Tabela 5 - Otimização dos filtros para o sinal de *HeaviSine* com $SNR=5$ e 10 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	30	31	3.23	1.7139	0.3832	7.15E-03	6.80E-03
EWMA	0.0641	0.0621	3.23	1.7141	0.3821	7.02E-03	7.65E-03
BWT	0.0302	0.0302	0	1.7161	0.3909	7.12E-03	7.02E-03
CBS	0.0201	0.0201	0	1.722	0.4161	7.08E-03	7.40E-03
ELI	0.0201	0.0201	0	1.7242	0.4265	7.08E-03	6.96E-02
OLMS	5	5	0	1.7304	0.4501	7.18E-03	6.10E-03
HF	0.1409	0.1409	0	1.8068	0.6452	2.71E-02	5.02E-02
	0.0609	0.0509	19.64				

Tabela 6 - Otimização dos filtros para o sinal de *HeaviSine* com $SNR=15$ e 10 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	13	13	0	0.5651	0.1896	2.28E-03	2.18E-03
EWMA	0.1423	0.1383	2.9	0.5628	0.1828	2.25E-03	2.06E-03
BWT	0.0762	0.0742	2.7	0.5679	0.1976	2.28E-03	2.28E-03
CBS	0.0542	0.0522	3.84	0.592	0.2596	2.18E-03	2.50E-03
ELI	0.0522	0.0502	4	0.5903	0.2562	2.18E-03	2.50E-03
OLMS	4	4	0	0.5671	0.1992	1.39E-03	1.77E-03
HF	0.3806	0.3606	5.55	0.6712	0.3665	2.34E-02	3.29E-02
	0.1009	0.1009	0				

Tabela 7 - Otimização dos filtros para o sinal de *HeaviSine* com $SNR=25$ e 10 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	5	5	0	0.1993	0.0943	7.59E-04	6.64E-04
EWMA	0.3307	0.3066	7.8	0.1967	0.0891	6.64E-04	7.59E-04
BWT	0.2225	0.2065	7.8	0.2032	0.1016	7.91E-04	6.96E-04
CBS	0.1624	0.1504	8	0.2659	0.2005	6.32E-04	7.91E-04
ELI	0.1564	0.1464	6.8	0.2643	0.1987	6.01E-04	7.91E-04
OLMS	3	3	0	0.1934	0.0818	6.32E-04	9.49E-04
HF	0.97	0.96	1.04	1.1746	0.9591	2.27E-01	1.95E-01
	0.1708	0.1908	10.48				

Tabela 8 - Otimização dos filtros para o sinal de *Doppler* com $SNR=5$ e 500 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	12	12	0	0.174	0.0586	7.60E-05	7.16E-05
EWMA	0.1363	0.1323	3.03	0.1728	0.0555	7.60E-05	6.71E-05
BWT	0.0843	0.0802	5	0.175	0.0614	8.05E-05	7.60E-05
CBS	0.0622	0.0602	3.33	0.1755	0.063	8.05E-05	7.60E-05
ELI	0.0622	0.0602	3.33	0.175	0.0617	8.05E-05	7.60E-05
OLMS	4	4	0	0.1762	0.0658	6.26E-05	7.16E-05
HF	0.8801 0.1608	0.8701 0.1508	1.15 6.63	0.5425	0.4247	1.92E-02	1.66E-02

Tabela 9 - Otimização dos filtros para o sinal de *Doppler* com $SNR=10$ e 500 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	7	8	12.5	0.1014	0.0402	4.47E-05	4.02E-05
EWMA	0.2204	0.2064	6.8	0.1007	0.0387	4.47E-05	4.02E-05
BWT	0.1424	0.1303	9.22	0.1024	0.0424	4.92E-04	4.47E-05
CBS	0.1043	0.0963	8.32	0.1033	0.0448	4.92E-05	4.92E-05
ELI	0.1043	0.0963	8.32	0.1027	0.0435	4.92E-05	4.47E-05
OLMS	3	3	0	0.1036	0.0446	5.81E-05	4.47E-05
HF	0.99 0.1708	0.99 0.1508	0 13.26	0.9067	0.7476	1.81E-02	1.50E-02

Tabela 10 - Otimização dos filtros para o sinal de *Doppler* com $SNR=25$ e 40 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	2	3	33.3	0.0227	0.0118	3.16E-05	3.16E-05
EWMA	0.6814	0.5531	23.2	0.0226	0.0116	3.16E-05	3.16E-05
BWT	0.6212	0.4429	40.3	0.0233	0.0125	4.74E-05	3.16E-05
CBS	0.487	0.3227	50.9	0.0282	0.0204	6.32E-05	6.32E-05
ELI	0.4449	0.3026	47	0.028	0.0201	3.16E-05	3.16E-05
OLMS	1	1	0	0.0228	0.013	3.16E-05	3.16E-05
HF	0.99 0.1708	0.99 0.1708	0 0	3.5328	2.947	3.19E-01	2.65E-01

Tabela 11 - Otimização dos filtros para o sinal de *Blocks* com $SNR=5$ e 250 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	8	8	0	1.2068	0.5017	9.68E-04	7.59E-04
EWMA	0.1964	0.1924	2.08	1.191	0.465	9.23E-04	7.08E-04
BWT	0.1243	0.1223	1.64	1.2225	0.5365	1.01E-03	8.10E-04
CBS	0.0923	0.0923	0	1.2286	0.5525	1.02E-03	8.85E-04
ELI	0.0903	0.0903	0	1.2201	0.5352	9.99E-04	8.79E-04
OLMS	4	4	0	1.1787	0.4323	7.02E-04	8.54E-04
HF	0.3506 0.1109	0.3207 0.1109	9.23 0	1.3258	0.641	8.03E-03	9.48E-03

Tabela 12 - Otimização dos filtros para o sinal de *Blocks* com SNR=10 e 250 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	4	5	20	0.7355	0.3575	6.45E-04	4.55E-04
EWMA	0.3246	0.3106	4.52	0.7202	0.3303	5.95E-04	4.11E-04
BWT	0.2345	0.2285	2.63	0.7535	0.389	6.77E-04	4.87E-04
CBS	0.1764	0.1724	2.32	0.7638	0.4119	6.83E-04	5.00E-04
ELI	0.1684	0.1644	2.44	0.7558	0.3991	6.64E-04	5.00E-04
OLMS	4	4	0	0.6896	0.267	5.57E-04	4.81E-04
HF	0.4006 0.0709	0.3806 0.0709	5.25 0	0.7655	0.3932	2.15E-03	2.54E-03

Tabela 13 - Otimização dos filtros para o sinal de *Blocks* com SNR=15 e 250 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	3	3	0	0.4696	0.2443	3.48E-04	2.72E-04
EWMA	0.511	0.4669	9.44	0.4564	0.2272	3.29E-04	2.59E-04
BWT	0.489	0.4289	14.02	0.4859	0.2703	3.92E-04	2.85E-04
CBS	0.4008	0.3407	17.64	0.5019	0.3021	3.86E-04	3.04E-04
ELI	0.3628	0.3187	13.83	0.497	0.2952	3.73E-04	3.10E-04
OLMS	3	4	25	0.4223	0.1567	3.16E-04	3.73E-04
HF	0.7203 0.1209	0.6703 0.1409	7.5 14.19	0.6149	0.4063	8.70E-03	1.01E-02

Tabela 14 - Otimização dos filtros para o sinal de *Blocks* com SNR=25 e 20 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	1	1	0	0.2233	0.1068	5.37E-04	3.35E-04
EWMA	1	0.7976	25.4	0.2233	0.0948	5.37E-04	2.68E-04
BWT	0.999	0.9037	10.6	0.2233	0.1039	5.37E-04	3.13E-04
CBS	0.999	0.9057	10.4	0.257	0.17	4.47E-04	4.47E-04
ELI	0.999	0.8857	12.9	0.257	0.1698	4.47E-04	4.47E-04
OLMS	3	4	25	0.2112	0.0512	4.02E-04	4.47E-04
HF	0.99 0.1608	0.96 0.1508	3.13 6.63	0.9765	0.7663	2.18E-01	1.88E-01

Tabela 15 - Otimização dos filtros para o sinal de *Bumps* com SNR=5 e 100 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	10	10	0	0.4042	0.1429	2.80E-04	1.80E-04
EWMA	0.1703	0.1623	4.94	0.4025	0.1385	2.80E-04	1.80E-04
BWT	0.1003	0.0943	6.38	0.4067	0.1491	2.90E-04	1.90E-04
CBS	0.0722	0.0682	5.87	0.4072	0.1509	2.90E-04	2.00E-04
ELI	0.0722	0.0682	5.87	0.4055	0.1471	2.90E-04	1.90E-04
OLMS	4	4	0	0.4059	0.1477	3.30E-04	2.30E-04
HF	0.4506 0.1409	0.4106 0.1409	9.74 0	0.5136	0.2879	1.12E-02	1.32E-02

Tabela 16 - Otimização dos filtros para o sinal de *Bumps* com $SNR=10$ e 100 iterações.

Filtros	$\hat{P}_{\acute{o}t\acute{i}m\acute{o}}$	$P_{\acute{o}t\acute{i}m\acute{o},r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	6	7	14.29	0.2363	0.097	1.60E-04	1.10E-04
EWMA	0.2605	0.2405	8.33	0.2353	0.0948	1.60E-04	1.00E-04
BWT	0.1584	0.1444	9.72	0.2387	0.102	1.70E-04	1.10E-04
CBS	0.1123	0.1023	9.79	0.2406	0.1068	1.70E-04	1.10E-04
ELI	0.1103	0.1003	9.99	0.239	0.1036	1.60E-04	1.10E-04
OLMS	3	4	25	0.2371	0.0984	2.10E-04	1.60E-04
HF	0.7303 0.1608	0.7003 0.1608	0 0	0.4366	0.3079	2.38E-02	2.20E-02

Tabela 17 - Otimização dos filtros para o sinal de *Bumps* com $SNR=15$ e 100 iterações.

Filtros	$\hat{P}_{\acute{o}t\acute{i}m\acute{o}}$	$P_{\acute{o}t\acute{i}m\acute{o},r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	4	5	20	0.1405	0.0652	9.00E-05	6.00E-05
EWMA	0.3828	0.3387	13.02	0.14	0.0641	9.00E-05	6.00E-05
BWT	0.2566	0.2205	16.36	0.1429	0.0691	9.00E-05	6.00E-05
CBS	0.1804	0.1564	15.37	0.1472	0.0782	9.00E-05	7.00E-05
ELI	0.1744	0.1504	16	0.1458	0.0759	9.00E-05	7.00E-05
OLMS	3	3	0	0.1403	0.0627	6.00E-05	5.00E-05
HF	0.8701 0.1608	0.8701 0.1708	0 5.86	0.357	0.2778	1.92E-02	1.69E-02

Tabela 18 - Otimização dos filtros para o sinal de *Bumps* com $SNR=25$ e 20 iterações.

Filtros	$\hat{P}_{\acute{o}t\acute{i}m\acute{o}}$	$P_{\acute{o}t\acute{i}m\acute{o},r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	2	2	0	0.0544	0.0283	6.71E-05	4.47E-05
EWMA	0.7375	0.5892	25.2	0.0539	0.0276	6.71E-05	4.47E-05
BWT	0.7234	0.495	46.1	0.0556	0.0301	8.94E-05	4.47E-05
CBS	0.6292	0.3668	71.6	0.068	0.0498	6.71E-05	6.71E-05
ELI	0.5611	0.3447	62.8	0.0677	0.0493	6.71E-05	6.71E-05
OLMS	2	2	0	0.0527	0.0248	8.94E-05	4.47E-05
HF	0.99 0.1708	0.99 0.1608	0 6.22	0.8767	0.7288	1.50E-01	1.27E-01

Tabela 19 - Otimização dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem, com $SNR=5$ e 40 iterações.

Filtros	$\hat{P}_{\acute{o}t\acute{i}m\acute{o}}$	$P_{\acute{o}t\acute{i}m\acute{o},r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	24	24	0	86.0968	21.803	1.49E-01	1.53E-01
EWMA	0.0762	0.0741	2.7	85.9702	21.197	1.50E-01	1.71E-01
BWT	0.0422	0.0422	0	86.4299	23.1009	1.50E-01	1.54E-01
CBS	0.0342	0.0342	0	92.6584	40.3456	1.41E-01	1.62E-01
ELI	0.0342	0.0342	0	92.6243	40.3155	1.35E-01	1.62E-01
OLMS	5	5	0	87.0403	24.3891	1.41E-01	2.00E-01
HF	0.0709 0.011	0.0709 0.011	0 0	86.0829	21.1071	7.74E-02	7.29E-02

Tabela 20 - Otimização dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem, com SNR=10 e 40 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	15	15	0	49.2905	15.2906	8.79E-02	9.40E-02
EWMA	0.1182	0.1162	1.72	49.1748	14.8068	8.77E-02	1.00E-01
BWT	0.0682	0.0682	0	49.6773	16.4477	8.90E-02	9.36E-02
CBS	0.0582	0.0582	0	59.6186	36.6086	7.74E-02	1.17E-01
ELI	0.0582	0.0582	0	59.4828	36.4153	7.52E-02	1.16E-01
OLMS	4	4	0	49.6752	16.8233	1.12E-01	9.39E-02
HF	0.1109 0.011	0.1109 0.011	0 0	48.9667	15.2091	7.53E-02	7.91E-02

Tabela 21 - Otimização dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem, com SNR=15 e 40 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	10	10	0	28.6624	10.6693	5.85E-02	5.39E-02
EWMA	0.1824	0.1743	4.6	28.5133	10.263	5.70E-02	4.93E-02
BWT	0.1103	0.1083	1.9	29.0819	11.6614	6.05E-02	5.57E-02
CBS	0.0963	0.0943	2.1	43.7379	34.5639	4.29E-02	7.67E-02
ELI	0.0943	0.0943	0	43.5974	34.4068	4.18E-02	7.70E-02
OLMS	4	4	0	28.8084	11.181	5.69E-02	4.42E-02
HF	0.1708 0.011	0.1708 0.011	0 0	28.9801	11.2097	4.89E-02	4.50E-02

Tabela 22 - Otimização dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem, com SNR=25 e 20 iterações.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
JM	3	4	25	10.3331	5.0501	2.97E-02	2.95E-02
EWMA	0.4148	0.3747	10.7	10.2041	4.8247	2.94E-02	2.84E-02
BWT	0.3407	0.2906	17.2	10.7074	5.73	2.90E-02	3.02E-02
CBS	0.3587	0.3067	17	34.0622	32.7535	1.02E-02	3.83E-02
ELI	0.3347	0.2866	16.8	34.0241	32.7133	9.39E-03	3.81E-02
OLMS	2	3	33.3	10.3144	5.1389	3.23E-02	2.68E-02
HF	0.4206 0.011	0.3806 0.021	10.51 47.62	11.2008	6.4219	2.32E-02	3.13E-02

Tabela 23 - Otimização dos filtros para o sinal resultante do sistema com overshoot e resposta inversa com SNR=5 e 10 iterações para EWMA e OLMS e uma iteração para KF.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
EWMA	0.2285	0.2224	2.7	0.6458	0.2697	0.0114	0.0077
OLMS	6	6	0	0.6467	0.2903	0.0135	0.0126
KF	7	7	0	-	-	-	-

Tabela 24 - Otimização dos filtros para o sinal resultante do sistema com *overshoot* e resposta inversa com $SNR=10$ e 10 iterações para EWMA e OLMS e uma iteração para KF.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
EWMA	0.3788	0.3687	2.7	0.3985	0.1918	0.0092	0.0054
OLMS	2	2	0	0.4068	0.2084	0.0113	0.005
KF	13	13	0	-	-	-	-

Tabela 25 - Otimização dos filtros para o sinal resultante do sistema com *overshoot* e resposta inversa com $SNR=15$ e 20 iterações para EWMA e OLMS e uma iteração para KF.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
EWMA	0.5551	0.5431	2.2	0.2641	0.1317	0.0033	0.0014
OLMS	2	2	0	0.2649	0.1393	0.0038	0.0012
KF	8	8	0	-	-	-	-

Tabela 26 - - Otimização dos filtros para o sinal resultante do sistema com *overshoot* e resposta inversa com $SNR=25$ e 20 iterações para EWMA e OLMS e uma iteração para KF.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
EWMA	0.8798	0.8778	0.22	0.1605	0.0524	0.0013	0.0009
OLMS	1	1	0	0.1568	0.0632	0.0012	0.0009
KF	5	5	0	-	-	-	-

Tabela 27 - Otimização do filtro de Kalman para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=5$ e uma iteração.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)
KF	10	10	0

Tabela 28 - Otimização do filtro de Kalman para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=10$ e uma iteração.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)
KF	10	9	11.11

Tabela 29 - Otimização do filtro de Kalman para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=15$ e uma iteração.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)
KF	12	12	0

Tabela 30 - Otimização do filtro de Kalman para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=25$ e uma iteração.

Filtros	$\hat{P}_{\acute{o}timo}$	$P_{\acute{o}timo,r}$	Desvio (%)
KF	9	9	0

Tabela 31 - Resultados da otimização do filtro de Kalman multiescala.

	$\hat{P}_{\text{ótimo}}$	$P_{\text{ótimo},r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	SNR
<i>HeaviSine</i>	5	5	0	1.7091	0.3631	5
	4	4	0	0.9565	0.2615	10
	1	1	0	0.5657	0.2049	15
	1	1	0	0.2013	0.1091	25
<i>Doppler</i>	4	4	0	0.1735	0.0593	5
	3	3	0	0.0989	0.0402	10
	2	2	0	0.0591	0.0299	15
	1	1	0	0.0217	0.014	25
<i>Blocks</i>	3	3	0	1.1724	0.546	5
	1	1	0	0.736	0.4217	10
	1	1	0	0.4752	0.3251	15
	1	1	0	0.2389	0.2109	25
<i>Bumps</i>	3	3	0	0.3999	0.1515	5
	3	3	0	0.2365	0.1119	10
	2	2	0	0.1418	0.0768	15
	1	1	0	0.052	0.0363	25
Resposta da junção de dois sistemas de primeira ordem	5	5	0	85.6674	20.5325	5
	4	4	0	50.2565	15.3733	10
	3	3	0	28.2776	10.9058	15
	2	2	0	10.1345	5.9364	25

Tabela 32 - Otimização do parâmetro dos filtros para o reator não isotérmico, com um SNR=5, 10 iterações para EWMA e OLMS e uma para MS-KF.

Filtros	$\hat{P}_{\text{ótimo}}$	$P_{\text{ótimo},r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
MS-KF	1	1	0	0.7262	0.5752	-	-
EWMA	0.485	0.6112	20.66	0.7245	0.3539	9.40E-04	5.10E-04
OLMS	2	1	100	0.7463	0.3852	9.60E-04	5.70E-04

Tabela 33 - Otimização do parâmetro dos filtros para o reator não isotérmico, com um SNR=10, 10 iterações para EWMA e OLMS e uma para MS-KF.

Filtros	$\hat{P}_{\text{ótimo}}$	$P_{\text{ótimo},r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
MS-KF	1	1	0	0.5694	0.5142	-	-
EWMA	0.6754	0.8136	17	0.5731	0.3539	5.40E-04	2.60E-04
OLMS	1	1	0	0.5858	0.2614	5.80E-04	2.80E-04

Tabela 34 - Otimização do parâmetro dos filtros para o reator não isotérmico, com um SNR=15, 10 iterações para EWMA e OLMS e uma para MS-KF.

Filtros	$\hat{P}_{\acute{o}t\text{imo}}$	$P_{\acute{o}t\text{imo},r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
MS-KF	1	1	0	0.5144	0.4929	-	-
EWMA	0.7876	0.9279	15.12	0.5165	0.1355	1.80E-04	1.80E-04
OLMS	1	1	0	0.5305	0.2156	2.40E-04	2.50E-04

Tabela 35 - Otimização do parâmetro dos filtros para o reator não isotérmico, com um SNR=15, 10 iterações para EWMA e OLMS e uma para MS-KF.

Filtros	$\hat{P}_{\acute{o}t\text{imo}}$	$P_{\acute{o}t\text{imo},r}$	Desvio (%)	\overline{RMSE}_{ap}	\overline{RMSE}_r	$\sigma_{\overline{RMSE}_{ap}}$	$\sigma_{\overline{RMSE}_r}$
MS-KF	1	1	0	0.484	0.831	-	-
EWMA	0.8557	0.992	13.74	0.4853	0.0445	1.00E-04	6.60E-05
OLMS	1	1	0	0.5056	0.2077	1.20E-04	6.70E-05

Anexo II

Resultados do teste dos filtros

Apresenta-se seguidamente todos os resultados decorrentes do teste dos filtros.

Tabela 36 - Resultados do teste dos filtros para o sinal de *HeaviSine* com *SNR*=10 e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.0096 <1E-9	-0.0088 <1E-15	-0.0495 <1E-12	-0.0487 <1E-12	-0.0275 <1E-11	-0.1819 <1E-24
EWMA	-0.0096 <1E-9	-	-0.0184 <1E-12	-0.0591 <1E-13	-0.0583 <1E-13	-0.0371 <1E-16	-0.1915 <1E-24
BWT	0.0088 <1E-15	0.0184 <1E-12	-	-0.0407 <1E-11	-0.0399 <1E-11	-0.0187 <1E-8	-0.173 <1E-23
CBS	0.0495 <1E-12	0.0591 <1E-13	0.0407 <1E-11	-	0.0008 <1E-3	0.022 <1E-6	-0.1323 <1E-18
ELI	0.0487 <1E-12	0.0583 <1E-13	0.0399 <1E-11	-0.0008 <1E-3	-	0.0212 <1E-5	-0.1332 <1E-18
OLMS	0.0275 <1E-11	0.0371 <1E-16	0.0187 <1E-8	-0.022 <1E-6	-0.0212 <1E-5	-	-0.1543 <1E-21
HF	0.1819 <1E-24	0.1915 <1E-24	0.173 <1E-23	0.1323 <1E-18	0.1332 <1E-18	0.1543 <1E-21	-

Tabela 37 - Resultados do teste dos filtros para o sinal de *HeaviSine* com *SNR*=15 e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.0075 <1E-11	-0.0083 <1E-19	-0.0702 <1E-18	-0.0667 <1E-18	-0.0035 0.021	-0.0848 <1E-21
EWMA	-0.0075 <1E-11	-	-0.0159 <1E-15	-0.0777 <1E-20	-0.0743 <1E-19	-0.0111 <1E-8	-0.0923 <1E-23
BWT	0.0083 <1E-19	0.0159 <1E-15	-	-0.0619 <1E-17	-0.0584 <1E-16	0.0048 0.0062	-0.0765 <1E-20
CBS	0.0702 <1E-18	0.0777 <1E-20	0.0619 <1E-17	-	0.0035 <1E-18	0.0667 <1E-18	-0.0146 <1E-5
ELI	0.0667 <1E-18	0.0743 <1E-19	0.0584 <1E-16	-0.0035 <1E-18	-	0.0632 <1E-17	-0.0181 <1E-6
OLMS	0.0035 0.021	0.0111 <1E-8	-0.0048 0.0062	-0.0667 <1E-18	-0.0632 <1E-17	-	-0.0813 <1E-21
HF	0.0848 <1E-21	0.0923 <1E-23	0.0765 <1E-20	0.0146 <1E-5	0.0181 <1E-6	0.0813 <1E-21	-

Tabela 38 - Resultados do teste dos filtros para o sinal de *HeaviSine* com $SNR=25$ e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.0053 <1E-19	-0.0079 <1E-26	-0.1082 <1E-29	-0.1064 <1E-28	0.0135 <1E-17	-0.0677 1E-29
EWMA	-0.0053 <1E-19	-	-0.0131 <1E-23	-0.1134 <1E-29	-0.1116 <1E-29	0.0082 <1E-14	-0.0729 <1E-31
BWT	0.0079 <1E-26	0.0131 <1E-23	-	-0.1003 <1E-28	-0.0985 <1E-27	0.0214 <1E-20	-0.0598 <1E-27
CBS	0.1082 <1E-29	0.1134 <1E-29	0.1003 <1E-28	-	0.0018 <1E-25	0.1216 <1E-28	0.0405 <1E-21
ELI	0.1064 <1E-28	0.1116 <1E-29	0.0985 <1E-27	-0.0018 <1E-25	-	0.1198 <1E-28	0.0387 <1E-20
OLMS	-0.0135 <1E-17	-0.0082 <1E-14	-0.0214 <1E-20	-0.1216 <1E-28	-0.1198 <1E-28	-	-0.0811 <1E-31
HF	0.0677 1E-29	0.0729 <1E-31	0.0598 <1E-27	-0.0405 <1E-21	-0.0387 <1E-20	0.0811 <1E-31	-

Tabela 39 - Resultados do teste dos filtros para o sinal de *Doppler* com $SNR=5$ e 150 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.0019 <5E-100	-0.0024 <5E-161	-0.0036 <5E-119	-0.0025 <1E-94	-0.0077 <1E-145	-0.0889 <1E-257
EWMA	-0.0019 <5E-100	-	-0.0043 <1E-130	-0.0055 <1E-124	-0.0044 <1E-109	-0.0096 <1E-160	-0.0908 <1E-260
BWT	0.0024 <5E-161	0.0043 <1E-130	-	-0.0012 <1E-58	-0.0001 0.0261	-0.0053 <1E-115	-0.0865 <1E-254
CBS	0.0036 <5E-119	0.0055 <1E-124	0.0012 <1E-58	-	0.0011 <1E-155	-0.0041 <1E-90	-0.0853 <1E-253
ELI	0.0025 <1E-94	0.0044 <1E-109	0.0001 0.0261	-0.0011 <1E-155	-	-0.0052 <1E-104	-0.0864 <1E-255
OLMS	0.0077 <1E-145	0.0096 <1E-160	0.0053 <1E-115	0.0041 <1E-90	0.0052 <1E-104	-	-0.0812 <1E-247
HF	0.0889 <1E-257	0.0908 <1E-260	0.0865 <1E-254	0.0853 <1E-253	0.0864 <1E-255	0.0812 <1E-247	-

Tabela 40 - Resultados do teste dos filtros para o sinal de *Doppler* com $SNR=10$ e 40 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.0012 <1E-28	-0.0017 <1E-47	-0.0039 <1E-36	-0.0029 <1E-30	-0.0043 <1E-37	-0.052 <1E-66
EWMA	-0.0012 <1E-28	-	-0.0029 <1E-38	-0.0051 <1E-37	-0.0041 <1E-33	-0.0055 <1E-42	-0.0532 <1E-66
BWT	0.0017 <1E-47	0.0029 <1E-38	-	-0.0022 <1E-27	-0.0012 <1E-17	-0.0026 <1E-28	-0.0503 <1E-64
CBS	0.0039 <1E-36	0.0051 <1E-37	0.0022 <1E-27	-	0.001 <1E-50	-0.0004 <1E-2	-0.0481 <1E-63
ELI	0.0029 <1E-30	0.0041 <1E-33	0.0012 <1E-17	-0.001 <1E-50	-	-0.0014 <1E-14	-0.0491 <1E-64
OLMS	0.0043 <1E-37	0.0055 <1E-42	0.0026 <1E-28	0.0004 <1E-2	0.0014 <1E-14	-	-0.0477 <1E-64
HF	0.052 <1E-66	0.0532 <1E-66	0.0503 <1E-64	0.0481 <1E-63	0.0491 <1E-64	0.0477 <1E-64	-

Tabela 41 - Resultados do teste dos filtros para o sinal de *Doppler* com SNR=15 e 40 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	3.8E-5 0.0112	-0.0020 <1E-49	-0.0058 <1E-47	-0.0005 <1E-44	-0.0051 <1E-48	-0.0253 <1E-67
EWMA	-3.8E-5 0.0112	-	-0.002 <1E-43	-0.0059 <1E-46	-0.0051 <1E-43	-0.0052 <1E-48	-0.0254 <1E-68
BWT	0.0020 <1E-49	0.002 <1E-43	-	-0.0039 <1E-40	-0.003 <1E-36	-0.0032 <1E-41	-0.0234 <1E-66
CBS	0.0058 <1E-47	0.0059 <1E-46	0.0039 <1E-40	-	0.0008 <1E-55	0.0007 <1E-12	-0.0195 <1E-60
ELI	0.0005 <1E-44	0.0051 <1E-43	0.003 <1E-36	-0.0008 <1E-55	-	-0.0001 <1E-2	-0.0203 <1E-61
OLMS	0.0051 <1E-48	0.0052 <1E-48	0.0032 <1E-41	-0.0007 <1E-12	0.0001 <1E-2	-	-0.0202 <1E-67
HF	0.0253 <1E-67	0.0254 <1E-68	0.0234 <1E-66	0.0195 <1E-60	0.0203 <1E-61	0.0202 <1E-67	-

Tabela 42 - Resultados do teste dos filtros para o sinal de *Doppler* com SNR=25 e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	-0.0001 <1E-11	-0.0012 <1E-29	-0.0091 <1E-33	-0.0088 <1E-32	-0.0011 <1E-24	-0.0043 <1E-29
EWMA	0.0001 <1E-11	-	-0.001 <1E-24	-0.0089 <1E-33	-0.0086 <1E-33	-0.0009 <1E-24	-0.0041 <1E-30
BWT	0.0012 <1E-29	0.001 <1E-24	-	-0.0079 <1E-31	-0.0076 <1E-30	0.0001 <1E-6	-0.0031 <1E-27
CBS	0.0091 <1E-33	0.0089 <1E-33	0.0079 <1E-31	-	0.0003 <1E-26	0.008 <1E-31	0.0048 <1E-25
ELI	0.0088 <1E-32	0.0086 <1E-33	0.0076 <1E-30	-0.0003 <1E-26	-	0.0077 <1E-30	0.0045 <1E-24
OLMS	0.0011 <1E-24	0.0009 <1E-24	-0.0001 <1E-6	-0.008 <1E-31	-0.0077 <1E-30	-	-0.0032 <1E-29
HF	0.0043 <1E-29	0.0041 <1E-30	0.0031 <1E-27	-0.0048 <1E-25	-0.0045 <1E-24	0.0032 <1E-29	-

Tabela 43 - Resultados do teste dos filtros para o sinal de *Blocks* com SNR=5 e 40 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.0387 <1E-35	-0.0365 <1E-49	-0.0527 <1E-36	-0.0347 <1E-29	0.0738 <1E-29	-0.0584 <1E-34
EWMA	-0.0387 <1E-35	-	-0.0752 <1E-41	-0.0913 <1E-38	-0.0733 <1E-34	0.0351 <1E-22	-0.0971 <1E-40
BWT	0.0365 <1E-49	0.0752 <1E-41	-	-0.0161 <1E-21	0.0019 0.0356	0.1103 <1E-34	-0.0219 <1E-17
CBS	0.0527 <1E-36	0.0913 <1E-38	0.0161 <1E-21	-	0.018 <1E-49	0.1264 <1E-33	-0.0058 0.0018
ELI	0.0347 <1E-29	0.0733 <1E-34	-0.0019 0.0356	-0.018 <1E-49	-	0.1084 <1E-31	-0.0238 <1E-15
OLMS	-0.0738 <1E-29	-0.0351 <1E-22	-0.1103 <1E-34	-0.1264 <1E-33	-0.1084 <1E-31	-	-0.1322 <1E-38
HF	0.0584 <1E-34	0.0971 <1E-40	0.0219 <1E-17	0.0058 0.0018	0.0238 <1E-15	0.1322 <1E-38	-

Tabela 44 - Resultados do teste dos filtros para o sinal de *Blocks* com SNR=10 e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.0277 <1E-20	-0.0316 <1E-24	-0.0545 <1E-21	-0.0412 <1E-18	0.0945 <1E-19	0.0084 <1E-10
EWMA	-0.0277 <1E-20	-	-0.0593 <1E-25	-0.0822 <1E-24	-0.069 <1E-22	0.0668 <1E-18	-0.0194 <1E-14
BWT	0.0316 <1E-24	0.0593 <1E-25	-	-0.0229 <1E-14	-0.0096 <1E-7	0.1261 <1E-21	0.0399 <1E-21
CBS	0.0545 <1E-21	0.0822 <1E-24	0.0229 <1E-14	-	0.0133 <1E-24	0.149 <1E-22	
ELI	0.0412 <1E-18	0.069 <1E-22	0.0096 <1E-7	-0.0133 <1E-24	-	0.1358 <1E-22	0.0496 <1E-18
OLMS	-0.0945 <1E-19	-0.0668 <1E-18	-0.1261 <1E-21	-0.149 <1E-22	-0.1358 <1E-22	-	-0.0862 <1E-19
HF	-0.0084 <1E-10	0.0194 <1E-14	-0.0399 <1E-21	-0.0628 <1E-20	-0.0496 <1E-18	0.0862 <1E-19	-

Tabela 45 - Resultados do teste dos filtros para o sinal de *Blocks* com SNR=15 e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.0158 <1E-21	-0.0265 <1E-22	-0.0603 <1E-24	-0.0531 <1E-23	0.0793 <1E-25	-0.0169 <1E-16
EWMA	-0.0158 <1E-21	-	-0.0423 <1E-24	-0.0761 <1E-25	-0.0689 <1E-25	0.0635 <1E-25	-0.0327 <1E-23
BWT	0.0265 <1E-22	0.0423 <1E-24	-	-0.0338 <1E-21	-0.0266 <1E-20	0.1058 <1E-25	0.0097 <1E-12
CBS	0.0603 <1E-24	0.0761 <1E-25	0.0338 <1E-21	-	0.0072 <1E-21	0.1396 <1E-27	0.0434 <1E-20
ELI	0.0531 <1E-23	0.0689 <1E-25	0.0266 <1E-20	-0.0072 <1E-21	-	0.1324 <1E-27	0.0362 <1E-19
OLMS	-0.0793 <1E-25	-0.0635 <1E-25	-0.1058 <1E-25	-0.1396 <1E-27	-0.1324 <1E-27	-	-0.0961 <1E-25
HF	0.0169 <1E-16	0.0327 <1E-23	-0.0097 <1E-12	-0.0434 <1E-20	-0.0362 <1E-19	0.0961 <1E-25	-

Tabela 46 - Resultados do teste dos filtros para o sinal de *Blocks* com SNR=25 e 150 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	1E-18 0.762	0.0001 <1E-35	-0.0675 <1E-243	-0.0675 <1E-243	0.0542 <1E-232	0.0011 <1E-207
EWMA	-1E-18 0.762	-	0.0001 <1E-35	-0.0675 <1E-243	-0.0675 <1E-243	0.0542 <1E-232	0.0011 <1E-207
BWT	-0.0001 <1E-35	-0.0001 <1E-35	-	-0.0676 <1E-243	-0.0676 <1E-243	0.0542 <1E-232	0.001 <1E-188
CBS	0.0675 <1E-243	0.0675 <1E-243	0.0676 <1E-243	-	1E-6 <1E-25	0.1217 <1E-268	0.0686 <1E-244
ELI	0.0675 <1E-243	0.0675 <1E-243	0.0676 <1E-243	-1E-6 <1E-25	-	0.1217 <1E-268	0.0686 <1E-244
OLMS	-0.0542 <1E-232	-0.0542 <1E-232	-0.0542 <1E-232	-0.1217 <1E-268	-0.1217 <1E-268	-	-0.0532 <1E-232
HF	-0.0011 <1E-207	-0.0011 <1E-207	-0.001 <1E-188	-0.0686 <1E-244	-0.0686 <1E-244	0.0532 <1E-232	-

Tabela 47 - Resultados do teste dos filtros para o sinal de *Bumps* com $SNR=5$ e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.0045 <1E-18	-0.0064 <1E-25	-0.0082 <1E-18	-0.0043 <1E-12	-0.003 <1E-8	-0.0703 <1E-29
EWMA	-0.0045 <1E-18	-	-0.0109 <1E-22	-0.0127 <1E-20	-0.0089 <1E-17	-0.0075 <1E-17	-0.0748 <1E-29
BWT	0.0064 <1E-25	0.0109 <1E-22	-	-0.0017 <1E-6	0.0021 <1E-7	0.0034 <1E-8	-0.0638 <1E-28
CBS	0.0082 <1E-18	0.0127 <1E-20	0.0017 <1E-6	-	0.0038 <1E-26	0.0051 <1E-10	-0.0621 <1E-29
ELI	0.0043 <1E-12	0.0089 <1E-17	-0.0021 <1E-7	-0.0038 <1E-26	-	0.0013 <1E-30	-0.0659 <1E-30
OLMS	0.003 <1E-8	0.0075 <1E-17	-0.0034 <1E-8	-0.0051 <1E-10	-0.0013 0.0031	-	-0.0672 <1E-27
HF	0.0703 <1E-29	0.0748 <1E-29	0.0638 <1E-28	0.0621 <1E-29	0.0659 <1E-30	0.0672 <1E-27	-

Tabela 48 - Resultados do teste dos filtros para o sinal de *Bumps* com $SNR=10$ e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.0027 <1E-18	-0.0046 <1E-30	-0.0099 <1E-23	-0.0066 <1E-19	-0.0017 <1E-11	-0.0638 <1E-35
EWMA	-0.0027 <1E-18	-	-0.0073 <1E-25	-0.0126 <1E-25	-0.0094 <1E-22	-0.0044 <1E-19	-0.0665 <1E-35
BWT	0.0046 <1E-30	0.0073 <1E-25	-	-0.0053 <1E-17	-0.0021 <1E-9	0.0029 <1E-14	-0.0592 <1E-34
CBS	0.0099 <1E-23	0.0126 <1E-25	0.0053 <1E-17	-	0.0032 <1E-31	0.0082 <1E-22	-0.0539 <1E-34
ELI	0.0066 <1E-19	0.0094 <1E-22	0.0021 <1E-9	-0.0032 <1E-31	-	0.0049 <1E-18	-0.0572 <1E-34
OLMS	0.0017 <1E-11	0.0044 <1E-19	-0.0029 <1E-14	-0.0082 <1E-22	-0.0049 <1E-18	-	-0.0621 <1E-36
HF	0.0638 <1E-35	0.0665 <1E-35	0.0592 <1E-34	0.0539 <1E-34	0.0572 <1E-34	0.0621 <1E-36	-

Tabela 49 - Resultados do teste dos filtros para o sinal de *Bumps* com $SNR=15$ e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.0012 <1E-20	-0.0039 <1E-31	-0.013 <1E-26	-0.0107 <1E-24	0.0037 <1E-19	-0.0382 <1E-33
EWMA	0.0012 <1E-20	-	-0.0051 <1E-27	-0.0142 <1E-27	-0.0119 <1E-25	0.0025 <1E-18	-0.0394 <1E-34
BWT	-0.0039 <1E-31	-0.0051 <1E-27	-	-0.009 <1E-23	-0.0068 <1E-21	0.0076 <1E-24	-0.0342 <1E-32
CBS	-0.013 <1E-26	-0.0142 <1E-27	-0.009 <1E-23	-	0.0022 <1E-35	0.0166 <1E-27	-0.0252 <1E-27
ELI	-0.0107 <1E-24	-0.0119 <1E-25	-0.0068 <1E-21	0.0022 <1E-35	-	0.0144 <1E-25	-0.0274 <1E-28
OLMS	0.0037 <1E-19	0.0025 <1E-18	0.0076 <1E-24	0.0166 <1E-27	0.0144 <1E-25	-	-0.0418 <1E-34
HF	-0.0382 <1E-33	-0.0394 <1E-34	-0.0342 <1E-32	-0.0252 <1E-27	-0.0274 <1E-28	-0.0418 <1E-34	-

Tabela 50 - Resultados do teste dos filtros para o sinal de *Bumps* com $SNR=25$ e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	-0.0011 <1E-24	-0.004 <1E-37	-0.0232 <1E-39	-0.0225 <1E-39	0.0033 <1E-29	-0.0088 <1E-34
EWMA	0.0011 <1E-24	-	-0.0028 <1E-33	-0.022 <1E-40	-0.0213 <1E-39	0.0045 <1E-31	-0.0076 <1E-37
BWT	0.004 <1E-37	0.0028 <1E-33	-	-0.0192 <1E-38	-0.0185 <1E-37	0.0073 <1E-35	-0.0048 <1E-31
CBS	0.0232 <1E-39	0.022 <1E-40	0.0192 <1E-38	-	0.0007 <1E-30	0.0265 <1E-41	0.0144 <1E-35
ELI	0.0225 <1E-39	0.0213 <1E-39	0.0185 <1E-37	-0.0007 <1E-30	-	0.0258 <1E-41	0.0137 <1E-35
OLMS	-0.0033 <1E-29	-0.0045 <1E-31	-0.0073 <1E-35	-0.0265 <1E-41	-0.0258 <1E-41	-	-0.0121 <1E-36
HF	0.0088 <1E-34	0.0076 <1E-37	0.0048 <1E-31	-0.0144 <1E-35	-0.0137 <1E-35	0.0121 <1E-36	-

Tabela 51 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=5$ e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.934 <1E-12	-1.0251 <1E-15	-19.3956 <1E-24	-19.4542 <1E-24	-1.9092 <1E-9	0.7848 <1E-7
EWMA	-0.934 <1E-12	-	-1.9591 <1E-15	-20.3296 <1E-25	-20.3882 <1E-25	-2.8432 <1E-14	-0.1492 0.02
BWT	1.0251 <1E-15	1.9591 <1E-15	-	-18.3705 <1E-24	-18.429 <1E-24	-0.8841 0.0001	1.8099 <1E-13
CBS	19.3956 <1E-24	20.3296 <1E-25	18.3705 <1E-24	-	-0.0585 <1E-5	17.4865 <1E-22	20.1804 <1E-23
ELI	19.4542 <1E-24	20.3882 <1E-25	18.429 <1E-24	0.0585 <1E-5	-	17.545 <1E-22	20.2389 <1E-23
OLMS	1.9092 <1E-9	2.8432 <1E-14	0.8841 0.0001	-17.4865 <1E-22	-17.545 <1E-22	-	2.694 <1E-10
HF	-0.7848 <1E-7	0.1492 0.02	-1.8099 <1E-13	-20.1804 <1E-23	-20.2389 <1E-23	-2.694 <1E-10	-

Tabela 52 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=10$ e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.6221 <1E-12	-0.6974 <1E-19	-22.6098 <1E-27	-22.478 <1E-27	-1.7156 <1E-15	0.6006 <1E-8
EWMA	-0.6221 <1E-12	-	-1.3195 <1E-16	-23.2319 <1E-27	-23.1001 <1E-27	-2.3377 <1E-17	-0.0215 0.001
BWT	0.6974 <1E-19	1.3195 <1E-16	-	-21.9125 <1E-26	-21.7807 <1E-26	-1.0182 <1E-11	1.298 <1E-13
CBS	22.6098 <1E-27	23.2319 <1E-27	21.9125 <1E-26	-	0.1318 <1E20	20.8943 <1E-26	23.2105 <1E-25
ELI	22.478 <1E-27	23.1001 <1E-27	21.7807 <1E-26	-0.1318 <1E20	-	20.7624 <1E-26	23.0786 <1E-25
OLMS	1.7156 <1E-15	2.3377 <1E-17	1.0182 <1E-11	-20.8943 <1E-26	-20.7624 <1E-26	-	2.3162 <1E-15
HF	-0.6006 <1E-8	0.0215 0.001	-1.298 <1E-13	-23.2105 <1E-25	-23.0786 <1E-25	-2.3162 <1E-15	-

Tabela 53 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com SNR=15 e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.3957 <1E-15	-0.6122 <1E-19	-24.772 <1E-34	-24.6419 <1E-34	-0.1264 0.0082	0.4901 <1E-12
EWMA	-0.3957 <1E-15	-	-1.008 <1E-19	-25.1677 <1E-35	-25.0376 <1E-34	-0.5222 <1E-10	0.0943 0.0017
BWT	0.6122 <1E-19	1.008 <1E-19	-	-24.1597 <1E-33	-24.0296 <1E-33	0.4858 <1E-7	1.1023 <1E-16
CBS	24.772 <1E-34	25.1677 <1E-35	24.1597 <1E-33	-	0.1301 <1E-21	24.6456 <1E-35	25.262 <1E-34
ELI	24.6419 <1E-34	25.0376 <1E-34	24.0296 <1E-33	-0.1301 <1E-21	-	24.5154 <1E-35	25.1319 <1E-34
OLMS	0.1264 0.0082	0.5222 <1E-10	-0.4858 <1E-7	-24.6456 <1E-35	-24.5154 <1E-35	-	0.6165 <1E-10
HF	-0.4901 <1E-12	-0.0943 0.0017	-1.1023 <1E-16	-25.262 <1E-34	-25.1319 <1E-34	-0.6165 <1E-10	-

Tabela 54 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com SNR=25 e 20 iterações.

	JM	EWMA	BWT	CBS	ELI	OLMS	HF
JM	-	0.403 <1E-106	-0.3393 <1E-123	-28.3009 <1E-223	-28.261 <1E-223	-0.0007 0.8953	0.4016 <1E-107
EWMA	-0.403 <1E-106	-	-0.7423 <1E-119	-28.7040 <1E-225	-28.664 <1E-225	-0.4038 <1E-90	-0.0015 0.01
BWT	0.3393 <1E-123	0.7423 <1E-119	-	-27.9617 <1E-222	-27.9217 <1E-222	0.3386 <1E-74	0.7409 <1E-121
CBS	28.3009 <1E-223	28.7040 <1E-225	27.9617 <1E-222	-	0.04 <1E-127	28.3002 <1E-224	28.7025 <1E-224
ELI	28.261 <1E-223	28.664 <1E-225	27.9217 <1E-222	-0.04 <1E-127	-	28.2603 <1E-224	28.6626 <1E-224
OLMS	0.0007 0.8953	0.4038 <1E-90	-0.3386 <1E-74	-28.3002 <1E-224	-28.2603 <1E-224	-	0.4023 <1E-89
HF	-0.4016 <1E-107	0.0015 0.01	-0.7409 <1E-121	-28.7025 <1E-224	-28.6626 <1E-224	-0.4023 <1E-89	-

Tabela 55 - Resultados do teste dos filtros para o sistema com resposta inversa e overshoot com SNR=5 e 10 iterações.

	KF	EWMA	OLMS
KF	-	-0.0399 <1E-9	-0.0858 <1E-8
EWMA	0.0399 <1E-9	-	-0.0459 <1E-6
OLMS	0.0858 <1E-8	0.0459 <1E-6	-

Tabela 56 - Resultados do teste dos filtros para o sistema com resposta inversa e overshoot com SNR=10 e 10 iterações.

	KF	EWMA	OLMS
KF	-	-0.0052 0.0009	-0.0289 <1E-8
EWMA	0.0052 0.0009	-	-0.0237 <1E-10
OLMS	0.0289 <1E-8	0.0237 <1E-10	-

Tabela 57 - Resultados do teste dos filtros para o sistema com resposta inversa e overshoot com SNR=25 e 10 iterações

	KF	EWMA	OLMS
KF	-	0.0188 <1E-9	0.0134 <1E-9
EWMA	-0.0188 <1E-9	-	-0.0054 <1E-6
OLMS	-0.0134 <1E-9	0.0054 <1E-6	-

Tabela 58 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com SNR=5 e 10 iterações.

	KF	EWMA	OLMS
KF	-	-4.498 <1E-9	-7.6622 <1E-11
EWMA	4.498 <1E-9	-	-3.1642 <1E-7
OLMS	7.6622 <1E-11	3.1642 <1E-7	-

Tabela 59 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com SNR=10 e 10 iterações.

	KF	EWMA	OLMS
KF	-	-2.7811 <1E-7	-4.9413 <1E-9
EWMA	2.7811 <1E-7	-	-2.1601 <1E-8
OLMS	4.9413 <1E-9	2.1601 <1E-8	-

Tabela 60 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com SNR=15 e 10 iterações.

	KF	EWMA	OLMS
KF	-	-3.575 <1E-12	-4.1859 <1E-14
EWMA	3.575 <1E-12	-	-0.6108 <1E-7
OLMS	4.1859 <1E-14	0.6108 <1E-7	-

Tabela 61 - Resultados do teste dos filtros para o sinal resultante da junção de dois sistemas de primeira ordem com $SNR=25$ e 10 iterações.

	KF	EWMA	OLMS
KF	-	-1.9418 <1E-13	-2.348 <1E-14
EWMA	1.9418 <1E-13	-	-0.4061 1E-7
OLMS	2.348 <1E-14	0.4061 1E-7	-

Tabela 62 - Resultados do teste dos filtros para o reator não isotérmico, com um $SNR=5$ e 10 iterações.

	MS-KF	EWMA	OLMS
MS-KF	-	0.196 <1E-14	0.14 <1E-13
EWMA	-0.196 <1E-14	-	-0.056 <1E-12
OLMS	-0.14 <1E-13	0.056 <1E-12	-

Tabela 63 - Resultados do teste dos filtros para o reator não isotérmico, com um $SNR=10$ e 10 iterações.

	MS-KF	EWMA	OLMS
MS-KF	-	0.2739 <1E-16	0.2524 <1E-16
EWMA	-0.2739 <1E-16	-	-0.0215 <1E-9
OLMS	-0.2524 <1E-16	0.0215 <1E-9	-

Tabela 64 - Resultados do teste dos filtros para o reator não isotérmico, com um $SNR=15$ e 10 iterações.

	MS-KF	EWMA	OLMS
MS-KF	-	0.3401 <1E-19	0.2769 <1E-16
EWMA	-0.3401 <1E-19	-	6.6869 <1E-15
OLMS	-0.2769 <1E-16	-6.6869 <1E-15	-

Tabela 65 - Resultados do teste dos filtros para o reator não isotérmico, com um SNR=25 e 10 iterações.

	MS-KF	EWMA	OLMS
MS-KF	-	0.406 <1E-22	0.277 <1E-20
EWMA	-0.406 <1E-22	-	-0.129 <1E-18
OLMS	-0.277 <1E-20	0.129 <1E-18	-

Anexo III

Códigos Matlab utilizados na otimização dos filtros

Nesta secção apresentam-se os códigos em Matlab utilizados para a otimização dos filtros.

```
clear
clc
close all
format short

%% Sinais em estudo %%

%Heavisine
%y_true=MakeSignal('HeaviSine',1000);

% Doppler
%y_true=MakeSignal('Doppler',5000);

% Blocks
%y_true=MakeSignal('Blocks',3000);

% Bumps
%y_true=MakeSignal('Bumps',15000);

% Resposta de um sistema de primeira ordem
%load('resposta_sistema_1ordem')

% Sinal multi-escala
%load('sinal_multi_escala')
%y_true=sinal_multi_escala;

% Geração do ruído
t=1:length(y_true);
SNR=5;
dp_ruído=sqrt(var(y_true)/10^(SNR/10));
it_max = 10;
ruído =randn(it_max,length(y_true));
k=1;

%-----%
% Otimização dos parametros de cada filtro%%
%-----%

%% Janela móvel online %%

m_jm_on_max=40;
it_jm_on_max = it_max;
RMSE_jm_on_re= zeros(m_jm_on_max,it_jm_on_max);
lag_error_meam = zeros(m_jm_on_max,it_jm_on_max);

for mon=1:m_jm_on_max
```

Anexo III

```
for j=1:it_jm_on_max;

    sinal = y_true+dp_ruido.*ruido(j,:);

    [jm_on,jm_on_com] = Moving_average_online(sinal,mon);

    RMSE_jm_on_re(mon,j) = rmse(y_true(mon+1:end),jm_on);
    lag_error_meam(mon,j) = rmse(sinal(mon+1+k:end),jm_on(1:end-k));

end
end

%Apresentação dos resultados %%

Media_RMSE_jm_on_r = mean(RMSE_jm_on_re,2);
Media_lag_error_mean = mean(lag_error_meam,2);

DP_RMSE_jm_on_r=std(RMSE_jm_on_re,0,2);
DP_lag_erro_mean=std(lag_error_meam,0,2);

ii=1:m_jm_on_max;
tab2=[ii' Media_RMSE_jm_on_r DP_RMSE_jm_on_r Media_lag_error_mean
DP_lag_erro_mean];
disp('
                                Janela móvel online')
disp('-----')
disp('      m      Media      DP      Media      DP ')
disp('      RMSE_r      RMSE_r      Lag_1      Lag_1 ')
disp('-----')
disp(' ')
disp(tab2)
disp(' ')
disp('-----')

min_error_jm_on_ap = min(Media_lag_error_mean)
index_min_error_jm_on_ap = find(Media_lag_error_mean==min_error_jm_on_ap);
m_jm_on_opt_ap = ii(index_min_error_jm_on_ap)

min_error_jm_on_r = min(Media_RMSE_jm_on_r)
index_min_error_jm_on_r = find(Media_RMSE_jm_on_r==min_error_jm_on_r);
m_jm_on_opt_r = ii(index_min_error_jm_on_r)

DP_lag_erro_mean_m_jm_on_opt_ap = DP_lag_erro_mean(index_min_error_jm_on_ap)
DP_RMSE_jm_on_r = DP_RMSE_jm_on_r(index_min_error_jm_on_r)

desvio_jm_on = abs(m_jm_on_opt_ap-m_jm_on_opt_r)*100/m_jm_on_opt_r;

disp('m óptimo aparente (%)')
disp(m_jm_on_opt_ap)
disp('Desvio relativo ao real (%)')
disp(desvio_jm_on)

[jm_on,jm_on_com] = Moving_average_online(sinal,m_jm_on_opt_ap);

figure (1)
plot(ii,Media_lag_error_mean,'g',m_jm_on_opt_ap,min_error_jm_on_ap,'og'...
     ,ii,Media_RMSE_jm_on_r,'b',m_jm_on_opt_r,min_error_jm_on_r,'ob')
title('Janela móvel online')
```

Anexo III

```
legend('RMSE aparente','m óptimo aparente','RMSE real','m óptimo aparente')
xlabel('parametro m')
ylabel('RMSE')

figure (2)
plot(ii,DP_lag_erro_mean,'g',ii,DP_RMSE_jm_on_r,'b')
title('Janela móvel online')
legend('DP aparente','DP real')
xlabel('Parametro m')
ylabel('Desvio Padrão')

green=[0,1.0000,0.4000];

figure (3)
plot(t,sinal,'r',t(m_jm_on_opt_ap+1:end),jm_on,'k')
hold on
plot(t,y_true,'color',green)
title('Janela móvel online')
legend('Sinal com ruído','Sinal filtrado utilizando o melhor parâmetro')
xlabel('Tempo')
ylabel('Sinal')

%-----%

%% EWMA %%

k=1;
efin=1;
esp=500; % Gama de lambdas
it_exp_max = it_max; % iterações em cada lambda
error_exp_ap = zeros(esp,it_exp_max);
error_exp_re = zeros(esp,it_exp_max);

a=linspace(0,1,esp);

for p_a=1:length(a)

    for it=1:it_exp_max

        sinal = y_true+dp_ruído*ruído(it,:);

        [f_exp,exp_com] = fil_exp(sinal,a(p_a),efin);

        error_exp_ap(p_a,it) = rmse(f_exp(1:end-k),sinal(efin+k:end));
        error_exp_re(p_a,it) = rmse(f_exp,y_true(efin:end));

    end
end

mean_error_exp_ap = mean(error_exp_ap,2);
mean_error_exp_re = mean(error_exp_re,2);

dp_error_exp_ap =std(error_exp_ap,0,2);
dp_error_exp_re =std(error_exp_re,0,2);

min_error_exp_ap = min(mean_error_exp_ap)
index_ap = find(mean_error_exp_ap==min_error_exp_ap);
a_min_ap = a(index_ap)
```

Anexo III

```
min_error_exp_re = min(mean_error_exp_re)
index_re = find(mean_error_exp_re==min_error_exp_re);
a_min_re = a(index_re)

dp_error_exp_ap_a_min_ap = dp_error_exp_ap(index_ap)
dp_error_exp_re_a_min_re = dp_error_exp_re(index_re)

%% Apresentação dos resultados %%

tab2=[a' mean_error_exp_re dp_error_exp_re mean_error_exp_ap
dp_error_exp_ap];
disp('
          Filtro exponencial')
disp('-----')
disp('   lambda   Media   DP   Media   DP   ')
disp('          RMSE_re  RMSE_re  Lag_error_ap  Lag_error_ap  ')
disp('-----')
disp(' ')
disp(tab2)
disp(' ')
disp('-----')

disp('lambda óptimo')
disp(a_min_ap)
desvio=abs(a_min_ap-a_min_re)*100/a_min_re;
disp('Desvio relativamente ao óptimo (%)')
disp(desvio)

figure(4)
plot(a,mean_error_exp_ap,'g',a_min_ap,min_error_exp_ap,'og',a,...
      mean_error_exp_re,'b',a_min_re,min_error_exp_re,'ob')
legend('RMSE aparente','mínimo aparente','RMSE real','minimo real')
title('Optimização de \lambda');xlabel('\lambda');ylabel('RMSE lag 1')

figure(5)
plot(a,dp_error_exp_ap,'g',a,dp_error_exp_re,'b')
legend('DP aparente','DP aparente')
xlabel('\lambda')
ylabel('Desvio padrão')

% Aplicação do melhor valor de lambda %
[f_exp,exp_com] = fil_exp(sinal,a_min_ap,efin);

figure (6)
plot(t(efin+k:end),sinal(efin+k:end),'r',t(efin+k:end),f_exp(1:end-k),'k')
hold on
plot(t,y_true,'color',green)
plot(t,y_true,'g',t(efin+k:end),f_exp(1:end-k),'k')
title('EWMA')
legend('Sinal com ruído','Sinal filtrado utilizando o melhor parâmetro')
xlabel('Tempo')
ylabel('Sinal')

%-----%

%% Filtro Butterworth %%

Wn=linspace(0.0001,0.9999,500);
dim_a=length(Wn);
```

Anexo III

```
it_but_max =it_max;
n=2;

RMSE_butter_re= zeros(dim_a,it_but_max);
lag_error_butter = zeros(dim_a,it_but_max);

for m=1:dim_a

    for j=1:it_but_max;

        sinal = y_true+dp_ruido.*ruido(j,:);

        [coef_b,coef_a] = butter(n,Wn(m));
        filt_butter=filter(coef_b,coef_a,sinal);

        RMSE_butter_re(m,j) = rmse(y_true,filt_butter);
        lag_error_butter(m,j) = rmse(sinal(1+k:end),filt_butter(1:end-k));

    end
end

%Apresentação dos resultados %%

Media_RMSE_butter_r = mean(RMSE_butter_re,2);
Media_lag_error_butter = mean(lag_error_butter,2);

DP_RMSE_butter_r=std(RMSE_butter_re,0,2);
DP_lag_error_butter=std(lag_error_butter,0,2);

tab2=[Wn' Media_RMSE_butter_r DP_RMSE_butter_r Media_lag_error_butter
DP_lag_error_butter];
disp('
                        Filtro Butterworth')
disp('-----')
disp('
Wn      Media      DP      Media      DP ')
disp('      RMSE_re  RMSE_re  Lag_Butter  Lag_Butter ')
disp('-----')
disp(' ')
disp(tab2)
disp(' ')
disp('-----')
disp('-----')

min_error_butter_ap = min(Media_lag_error_butter)
index_min_error_butter_ap =
find(Media_lag_error_butter==min_error_butter_ap);
Wn_optimo_ap = Wn(index_min_error_butter_ap)

min_error_butter_r = min(Media_RMSE_butter_r)
index_min_error_butter_r = find(Media_RMSE_butter_r==min_error_butter_r);
Wn_optimo_r = Wn(index_min_error_butter_r)

DP_lag_error_butter_Wn_optimo_ap =
DP_lag_error_butter(index_min_error_butter_ap)
DP_RMSE_butter_r_Wn_optimo_r = DP_RMSE_butter_r(index_min_error_butter_r)

desvio_butter = abs(Wn_optimo_ap-Wn_optimo_r)*100/Wn_optimo_r;
```

Anexo III

```
disp('Frequência de corte óptima aparente')
disp(Wn_optimo_ap)
disp('Desvio relativo ao real (%)')
disp(desvio_butter)

[coef_b,coef_a] = butter(n,Wn_optimo_ap);
filt_butter=filter(coef_b,coef_a,sinal);

figure (7)
plot(Wn,Media_RMSE_butter_r,'b',Wn_optimo_r,min_error_butter_r,'ob',...
      Wn,Media_lag_error_butter,'g',Wn_optimo_ap,min_error_butter_ap,'og')
title('Filtro Butterworth')
legend('RMSE real','mínimo real','RMSE aparente','mínimo aparente')
xlabel('Wn')
ylabel('RMSE')

figure (8)
plot(Wn,DP_RMSE_butter_r,'b',Wn,DP_lag_error_butter,'g')
title('Filtro Butterworth')
legend('Dp real','Dp aparente')
xlabel('Wn')
ylabel('Desvio padrão')

figure (9)
plot(t,sinal,'r',t,filt_butter,'k')
hold on
plot(t,y_true,'color',green)
title('Filtro Butterworth')
legend('Sinal com ruído','Sinal filtrado utilizando o melhor parâmetro')
xlabel('Tempo')
ylabel('Sinal')

%% Filtro Chebyshev %%

Wp=linspace(0.0001,0.9999,500);
R=0.5;
dim_a=length(Wp);
it_cheby_max =it_max;
n=2;

RMSE_cheby_re= zeros(dim_a,it_cheby_max);
lag_error_cheby = zeros(dim_a,it_cheby_max);

for m=1:dim_a

    for j=1:it_cheby_max;

        sinal = y_true+dp_ruído.*ruído(j,:);

        [coef_b_cheb,coef_a_cheb] = cheby1(n,R,Wp(m));
        filt_cheby=filter(coef_b_cheb,coef_a_cheb,sinal);

        RMSE_cheby_re(m,j) = rmse(y_true,filt_cheby);
        lag_error_cheby(m,j) = rmse(sinal(1+k:end),filt_cheby(1:end-k));

    end
end
end
```

Anexo III

```
%Apresentação dos resultados %%

Media_RMSE_cheby_r = mean(RMSE_cheby_re,2);
Media_lag_error_cheby = mean(lag_error_cheby,2);

DP_RMSE_cheby_r=std(RMSE_cheby_re,0,2);
DP_lag_error_cheby=std(lag_error_cheby,0,2);

tab2=[Wp' Media_RMSE_cheby_r DP_RMSE_cheby_r Media_lag_error_cheby
DP_lag_error_cheby];
disp('
                                Filtro Chebyshev')
disp('-----')
disp('--')
disp('
      Wp          Media          DP          Media          DP ')
disp('
          RMSE_re   RMSE_re   Lag_cheby   Lag_cheby ')
disp('-----')
disp('--')
disp(' ')
disp(tab2)
disp(' ')
disp('-----')
disp('--')

min_error_cheby_ap = min(Media_lag_error_cheby)
index_min_error_cheby_ap = find(Media_lag_error_cheby==min_error_cheby_ap);
Wp_optimo_ap = Wp(index_min_error_cheby_ap)

min_error_cheby_r = min(Media_RMSE_cheby_r)
index_min_error_cheby_r = find(Media_RMSE_cheby_r==min_error_cheby_r);
Wp_optimo_r = Wp(index_min_error_cheby_r)

DP_lag_error_cheby_Wp_optimo_ap =
DP_lag_error_cheby(index_min_error_cheby_ap)
DP_RMSE_cheby_r_Wp_optimo_r = DP_RMSE_cheby_r(index_min_error_cheby_r)

desvio_cheby = abs(Wp_optimo_ap-Wp_optimo_r)*100/Wp_optimo_r;

disp('Frequência de corte óptima aparente')
disp(Wp_optimo_ap)
disp('Desvio relativo ao real (%)')
disp(desvio_cheby)

[coef_b_cheb,coef_a_cheb] = cheby1(n,R,Wp_optimo_ap);
filt_cheby=filter(coef_b_cheb,coef_a_cheb,sinal);

figure (10)
plot(Wp,Media_RMSE_cheby_r,'b',Wp_optimo_r,min_error_cheby_r,'ob',...
      Wp,Media_lag_error_cheby,'g',Wp_optimo_ap,min_error_cheby_ap,'og')
title('Filtro Chebyshev')
legend('RMSE real','mínimo real','RMSE aparente','mínimo aparente')
xlabel('Wp')
ylabel('RMSE')

figure (11)
plot(Wp,DP_RMSE_cheby_r,'b',Wp,DP_lag_error_cheby,'g')
title('Filtro Chebyshev')
legend('Dp real','Dp aparente')
```


Anexo III

```
xlabel('Wp')
ylabel('Desvio padrão')

figure (12)
plot(t,sinal,'g',t,filt_cheby,'b')
title('Filtro Chebyshev')
legend('Sinal com ruído','Sinal filtrado utilizando o melhor parâmetro')
xlabel('Tempo')
ylabel('Sinal')

%% Filtro Eliptico %%

Wp_elipt=linspace(0.0001,0.9999,500);
Rp = 0.5;
Rs=20;
n=2;
dim_a=length(Wp_elipt);
it_elipt_max =it_max;

RMSE_elipt_re= zeros(dim_a,it_elipt_max);
lag_error_elipt = zeros(dim_a,it_elipt_max);

for m=1:dim_a

    for j=1:it_elipt_max;

        sinal = y_true+dp_ruído.*ruído(j,:);

        [coef_b_elipt,coef_a_elipt] = ellip(n,Rp,Rs,Wp_elipt(m));
        filt_elipt=filter(coef_b_elipt,coef_a_elipt,sinal);

        RMSE_elipt_re(m,j) = rmse(y_true,filt_elipt);
        lag_error_elipt(m,j) = rmse(sinal(1+k:end),filt_elipt(1:end-k));

    end
end

%Apresentação dos resultados %%

Media_RMSE_elipt_r = mean(RMSE_elipt_re,2);
Media_lag_error_elipt = mean(lag_error_elipt,2);

DP_RMSE_elipt_r=std(RMSE_elipt_re,0,2);
DP_lag_error_elipt=std(lag_error_elipt,0,2);

tab2=[Wp_elipt' Media_RMSE_elipt_r DP_RMSE_elipt_r Media_lag_error_elipt
DP_lag_error_elipt];
disp('
                                Filtro Eliptico')
disp('-----')
disp('--')
disp('      Wp      Media      DP      Media      DP ')
disp('      RMSE_re  RMSE_re  Lag_elipt  Lag_elipt  ')
disp('-----')
disp('--')
disp(' ')
disp(tab2)
```

Anexo III

```
disp(' ')
disp('-----')
disp('--')

min_error_elipt_ap = min(Media_lag_error_elipt)
index_min_error_elipt_ap = find(Media_lag_error_elipt==min_error_elipt_ap);
Wp_elipt_optimo_ap = Wp_elipt(index_min_error_elipt_ap)

min_error_elipt_r = min(Media_RMSE_elipt_r)
index_min_error_elipt_r = find(Media_RMSE_elipt_r==min_error_elipt_r);
Wn_elipt_optimo_r = Wp_elipt(index_min_error_elipt_r)

DP_lag_error_elipt_Wp_elipt_optimo_ap =
DP_lag_error_elipt(index_min_error_elipt_ap)
DP_RMSE_elipt_r_Wn_elipt_optimo_r = DP_RMSE_elipt_r(index_min_error_elipt_r)

desvio_elipt = abs(Wp_elipt_optimo_ap-
Wn_elipt_optimo_r)*100/Wn_elipt_optimo_r;

disp('Frequência de corte óptima aparente')
disp(Wp_elipt_optimo_ap)
disp('Desvio relativo ao real (%)')
disp(desvio_elipt)

[coef_b_elipt,coef_a_elipt] = ellip(n,Rp,Rs,Wp_elipt_optimo_ap);
filt_elipt=filter(coef_b_elipt,coef_a_elipt,sinal);

figure (13)
plot(Wp_elipt,Media_RMSE_elipt_r,'b',Wn_elipt_optimo_r,min_error_elipt_r,'ob'
,...
Wp_elipt,Media_lag_error_elipt,'g',Wp_elipt_optimo_ap,min_error_elipt_ap,'og'
)
title('Filtro Eliptico')
legend('RMSE real','mínimo real','RMSE aparente','mínimo aparente')
xlabel('Wn')
ylabel('RMSE')

figure (14)
plot(Wp_elipt,DP_RMSE_elipt_r,'b',Wp_elipt,DP_lag_error_elipt,'g')
title('Filtro Eliptico')
legend('Dp real','Dp aparente')
xlabel('Wn')
ylabel('Desvio padrão')

figure (15)
plot(t,sinal,'g',t,filt_elipt,'b')
title('Filtro Eliptico')
legend('Sinal com ruído','Sinal filtrado utilizando o melhor parâmetro')
xlabel('Tempo')
ylabel('Sinal')

%% Filtro Wavelet online%%

Jdec_max=6;
it_wavelet_on_max = it_max;
RMSE_wavelet_on_re= zeros(Jdec_max,it_wavelet_on_max);
lag_error_wavelet = zeros(Jdec_max,it_wavelet_on_max);
```

Anexo III

```
for Jdec=1:Jdec_max
    for j=1:it_wavelet_on_max;
        sinal = y_true+dp_ruido.*ruido(j,:);
        wavelet_on = Wavelet_Online_Filter(sinal,Jdec);
        RMSE_wavelet_on_re(Jdec,j) =
rmse(y_true(2^Jdec:end),wavelet_on(2^Jdec:end)');
        lag_error_wavelet(Jdec,j) =
rmse(sinal(2^Jdec+k:end),wavelet_on(2^Jdec:end-k)');
    end
end

%Apresentação dos resultados %%

Media_RMSE_wavelet_on_r = mean(RMSE_wavelet_on_re,2);
Media_lag_error_wavelet = mean(lag_error_wavelet,2);

DP_RMSE_wavelet_on_r=std(RMSE_wavelet_on_re,0,2);
DP_lag_erro_wavelet=std(lag_error_wavelet,0,2);

ii=1:Jdec_max;
tab2=[ii' Media_RMSE_wavelet_on_r DP_RMSE_wavelet_on_r
Media_lag_error_wavelet DP_lag_erro_wavelet];
disp('
Wavelet online')
disp('-----')
disp('--')
disp('
Jdec      Media      DP      Media      DP ')
disp('
RMSE_r    RMSE_r    Lag_1    Lag_1 ')
disp('-----')
disp('--')
disp(' ')
disp(tab2)
disp(' ')
disp('-----')
disp('--')

min_error_wavelet_on_ap = min(Media_lag_error_wavelet)
index_min_error_wavelet_on_ap =
find(Media_lag_error_wavelet==min_error_wavelet_on_ap);
jdec_opt_ap = ii(index_min_error_wavelet_on_ap)

min_error_wavelet_on_r = min(Media_RMSE_wavelet_on_r)
index_min_error_wavelet_on_r =
find(Media_RMSE_wavelet_on_r==min_error_wavelet_on_r);
jdec_opt_r = ii(index_min_error_wavelet_on_r)

DP_lag_erro_jdec_jdec_opt_ap =
DP_lag_erro_wavelet(index_min_error_wavelet_on_ap)
DP_RMSE_wavelet_jdec_opt_r=
DP_RMSE_wavelet_on_r(index_min_error_wavelet_on_r)

desvio_wavelet = abs(jdec_opt_ap-jdec_opt_r)*100/jdec_opt_r;
```

Anexo III

```
disp('jdec óptimo aparente (%)')
disp(jdec_opt_ap)
disp('Desvio relativo ao real (%)')
disp(desvio_wavelet)

wavelet_on = Wavelet_Online_Filter(sinal,jdec_opt_ap);

figure (16)
plot(ii,Media_lag_error_wavelet,'g',jdec_opt_ap,min_error_wavelet_on_ap,'og'.
    ..
    ,ii,Media_RMSE_wavelet_on_r,'b',jdec_opt_r,min_error_wavelet_on_r,'ob')
title('Wavelet online')
legend('RMSE aparente','Jdec óptimo aparente','RMSE real','Jdec óptimo
aparente')
xlabel('parametro Jdec')
ylabel('RMSE')

figure (17)
plot(ii,DP_lag_erro_wavelet,'g',ii,DP_RMSE_wavelet_on_r,'b')
title('Wavelet online')
legend('DP aparente','DP real')
xlabel('Parametro Jdec')
ylabel('Desvio Padrão')

figure (18)
plot(t,sinal,'r',t(jdec_opt_ap:end),wavelet_on(jdec_opt_ap:end),'k')
hold on
plot(t,y_true,'color',green)
title('Wavelet online')
legend('Sinal com ruído','Sinal filtrado utilizando o melhor parâmetro')
xlabel('Tempo')
ylabel('Sinal')

%%
function [jm_on,jm_on_com] = Moving_average_online(sinal,m)

    % Filtro da média da janela móvel online

dim = length(sinal);
filt = zeros(1,dim);

for k=1:dim
    if k<=m
        filt(k)=sum(sinal(1:k))/k;
    else
        filt(k) = filt(k-1)+(sinal(k)-sinal(k-m))/m;
    end
end

jm_on_com = filt;
jm_on = filt(m+1:dim);

end

%%
function [exp,exp_com] = fil_exp(sinal,a,efin)

% Filtro exponencial
```

Anexo III

```
dim = length(sinal);
filt = zeros(1,dim);
filt(1)=sinal(1);

for k=2:dim
    filt(k) = a*sinal(k)+(1-a)*filt(k-1);
end

exp_com=filt;
exp=filt(efin:dim);

end

%%

function [sinal_filtrado,previsao_1_passo] = Hot_Winters(sinal,alfa2,beta,m)

% Filtro de Holt

sinal_filtrado=zeros(1,length(sinal));
previsao_1_passo=zeros(1,length(sinal));
b=zeros(1,length(sinal));

P=polyfit(sinal(1:m),1:m,1);
sinal_filtrado(m)=P(2)+P(1)*m;
b(m)=P(1);

for i=m+1:length(sinal)
    sinal_filtrado(i)=alfa2*sinal(i)+(1-alfa2)*(sinal_filtrado(i-1)+b(i-1));
    b(i)=beta*(sinal_filtrado(i)-sinal_filtrado(i-1))+ (1-beta)*b(i-1);
    previsao_1_passo(i)=sinal_filtrado(i-1)+b(i-1);
end

previsao_1_passo=previsao_1_passo(1:length(sinal));

end

%%

function [wavelet_filt,treshold_save] = Wavelet_Online_Filter(sinal,Jdec)

% Filtro multiescala online - OLMS

dim=length(sinal);
qmf = MakeONFilter('Haar',2);
wavelet_filt = zeros(dim,1);

for i=2^Jdec:dim

    x2L = 2^floor(log2(i));
    indx2 = [i-x2L+1:i]';
    Jmax = log2(length((indx2)));
```

Anexo III

```
L=Jmax-Jdec;
wc = FWT_PO(sinal(indx2),L,qmf);

sigma = zeros(Jdec,1);
treshold = zeros(Jdec,1);

for J=1:Jdec

    L2 = Jmax-J;
    index=dyad(L2);

    sigma(J) = median(abs(wc(index)))/0.6745;
    treshold(J) = sigma(J)*sqrt(2*log10(2^Jdec));

    treshold_save(i,J)=treshold(J);

    for k=index(1):index(end)

        if abs(wc(k))<=treshold(J)

            wc(k)=0;

        else
            wc(k) = sign(wc(k))*(abs(wc(k))-treshold(J));
        end

    end

end

%%

%% Kalman filter
sinal=(y_true(1:(1/2)*length(y_true))+dp_ruido*ruido(1,1:(1/2)*length(y_true)
));
sinal2=(y_true((1/2)*length(y_true):end)+dp_ruido*ruido(1,(1/2)*length(y_true)
):end));

data = iddata(sinal,ut(1:(1/2)*length(y_true)),1);
data2 = iddata(sinal2,ut((1/2)*length(y_true):end),1);

for nx_it=1:15
    SYS = n4sid(data,nx_it,'Ts',1);
    [YH, FIT(nx_it), X0]=compare(data2,SYS,1);
    RMSE_r(nx_it)=rmse(YH.y',y_true((1/2)*length(y_true):end));
    RMSE_ap(nx_it)=rmse(YH.y,sinal((1/2)*length(y_true):end));
end

max_FIT=max(FIT);
nx=find(FIT==max_FIT);
SYS = n4sid(data,nx,'Ts',1);

min_RMSE_r=min(RMSE_r);
nx_r=find(RMSE_r==min_RMSE_r);
erro_nx=abs(nx_r-nx)*100/nx_r;
disp('Erro real')
```

Anexo III

```
disp(RMSE_r(nx))
disp('Erro aparente')
disp(RMSE_ap(nx))
disp('Diferença entre ordens')
disp(erro_nx)

figure(1)
plot(FIT)
xlabel('nx')
ylabel('Fit to validation data')

figure(2)
compare(data2,SYS,1);

%%

% Filtro de Kalman multiescala

sinal=y_true+dp_ruido*ruido(1,:);
val_sinal=y_true+dp_ruido*ruido(2,:);
Jdec_max=6;

for Jdec=1:Jdec_max

coefficients = get_coeficiente(sinal,Jdec)';
coefficients_val= get_coeficiente(val_sinal,Jdec)';
%coefficients_input=get_coeficiente(entrada(1,:),Jdec)';

    clear YH_out
    clear FIT
for j=1:Jdec+1

    data = iddata(coefficients(:,j),[],1);
    val_data=iddata(coefficients_val(:,j),[],1);

    th = n4sid(data,'best','Ts',1);
    [YH, FIT(j), X0] = compare(val_data,th,1);
    YH_out(:,j) = YH.y;

    if FIT(j)<70;
        e=coefficients(:,j);
        for i=1:size(coefficients(:,j),1)
            YH_out(i,j)=th.C*th.K*coefficients(i,j);
        end
        disp('Ajuste abaixo do treshold')
        disp(FIT(:,j))
        end

        eval(['X0' num2str(j) ' = X0'])
        eval(['A' num2str(j) ' = th.A']);
        eval(['C' num2str(j) ' = th.C']);
        eval(['K' num2str(j) ' = th.K']);

end

wc=zeros(2^Jdec,1);
```

Anexo III

```
qmf=MakeONFilter('Haar',2);
L=0;
filt=zeros(dim,1);

for i=1:length(YH_out(:,1))

    for j=1:Jdec;
        pseud_coef=zeros(2^(Jdec-j),1);
        pseud_coef(end)=YH_out(i,j);
        wc(2^(Jdec-j):2^(Jdec-j+1)-1,1)=pseud_coef;
    end
    wc(1)=YH_out(i,end);

    pseud_filt=IWT_PO(wc,L,qmf);
    filt(i+2^Jdec-1)=pseud_filt(end);
end

RMSE_ap(Jdec)=rmse(filt(200:end),sinal(200:end));
RMSE_r(Jdec)=rmse(filt(200:end),y_true(200:end));

end

figure(1)
plot(RMSE_ap)
title('RMSE_ap')

figure(2)
plot(RMSE_r)
title('RMSE_r')

figure(3)
plot(val_sinal,'r')
hold on
plot(filt,'b')
hold on
plot(y_true,'g')
```


Anexo IV

Códigos Matlab utilizados no teste dos filtros

```
% Geração do ruído
x=1:length(y_true);
SNR=25;
dp_ruído=sqrt(var(y_true)/10^(SNR/10));
it_max = 10;
margem=200;
ruído =randn(it_max,length(y_true));

k=1;

%% Filtros%%

%% Janela móvel online %%

mon=3;
RMSE_jm_on_re = zeros(1,it_max);
lag_error_meam= zeros(1,it_max);

for it=1:it_max;

    sinal = y_true+dp_ruído.*ruído(it,:);

    [jm_on,jm_on_com] = Moving_average_online(sinal,mon);

    RMSE_jm_on_re(it) = rmse(jm_on_com(margem:end),y_true(margem:end));
    lag_error_meam(it) = rmse(sinal(mon+1+k:end),jm_on(1:end-k));

    jm_on(it,:)=jm_on;
end

Media_RMSE_jm_on_r = mean(RMSE_jm_on_re');
Media_lag_error_mean = mean(lag_error_meam');

DP_RMSE_jm_on_r=std(RMSE_jm_on_re,0,2);
DP_lag_erro_mean=std(lag_error_meam,0,2);

figure(1)
plot(x,sinal,'g',x(mon+1:end),jm_on(1,:),'b')
title('Janela móvel online')
legend('Sinal com ruído','Sinal filtrado')
xlabel('Tempo')
ylabel('Sinal')

%% Filtro exponencial %%

a= 0.7214;
efin =1;
```

Anexo IV

```
RMSE_exp_re= zeros(1,it_max);
lag_error_exp = zeros(1,it_max);

for j=1:it_max;

    sinal = y_true+dp_ruido.*ruido(j,:);

    [exp,exp_com] = fil_exp(sinal,a,efin);

    RMSE_exp_re(j) = rmse(y_true(margem:end),exp_com(margem:end));
    lag_error_exp(j) = rmse(exp(1:end-k),sinal(efin+k:end));
end

Media_RMSE_exp_r = mean(RMSE_exp_re');
Media_lag_error_exp = mean(lag_error_exp');

DP_RMSE_exp_r=std(RMSE_exp_re,0,2);
DP_lag_erro_exp=std(lag_error_exp,0,2);

green=[0,1.0000,0.4000];
figure(2)
plot(x,sinal,'r',x(efin:end),exp,'k');
hold on
plot(x,y_true,'color',green)
title('EWMA')
legend('Sinal com ruído','Sinal filtrado')
xlabel('Tempo')
ylabel('Sinal')

%% Filtro Butterworth %%

Wn= 0.3407;
n=2;

RMSE_butter_re= zeros(1,it_max);
lag_error_butter = zeros(1,it_max);
[coef_b,coef_a] = butter(n,Wn);

for it=1:it_max;

    sinal = y_true+dp_ruido.*ruido(it,:);

    filt_butter=filter(coef_b,coef_a,sinal);

    RMSE_butter_re(it) = rmse(y_true(margem:end),filt_butter(margem:end));
    lag_error_butter(it) = rmse(sinal(1+k:end),filt_butter(1:end-k));

end

Media_RMSE_butter_r = mean(RMSE_butter_re');
Media_lag_error_butter = mean(lag_error_butter');

DP_RMSE_butter_r=std(RMSE_butter_re,0,2);
DP_lag_erro_butter=std(lag_error_butter,0,2);

figure(3)
plot(x,sinal,'r',x,filt_butter,'k');
hold on
```

Anexo IV

```
plot(x,y_true,'color',green)
title('Filtro Butterworth')
legend('Sinal com ruído','Sinal filtrado')
xlabel('Tempo')
ylabel('Sinal')

%% Filtro Chebyshev %%

Wp= 0.3587;
R=0.5;
n=2;

RMSE_cheby_re= zeros(1,it_max);
lag_error_cheby = zeros(1,it_max);
[coef_b_cheby,coef_a_cheby] = cheby1(n,R,Wp);

for it=1:it_max;

    sinal = y_true+dp_ruído.*ruído(it,:);

    filt_cheby=filter(coef_b_cheby,coef_a_cheby,sinal);

    RMSE_cheby_re(it) = rmse(y_true(margem:end),filt_cheby(margem:end));
    lag_error_cheby(it) = rmse(sinal(1+k:end),filt_cheby(1:end-k));

end

Media_RMSE_cheby_r = mean(RMSE_cheby_re');
Media_lag_error_cheby = mean(lag_error_cheby');

DP_RMSE_cheby_r=std(RMSE_cheby_re,0,2);
DP_lag_erro_cheby=std(lag_error_cheby,0,2);

figure(4)
plot(x,sinal,'g',x,filt_cheby,'b');
title('Filtro Chebyshev')
legend('Sinal com ruído','Sinal filtrado')
xlabel('Tempo')
ylabel('Sinal')

%% Filtro Eliptico %%

Wp_elipt= 0.3347;
Rp=0.5;
Rs=20;
n=2;

RMSE_elipt_re= zeros(1,it_max);
lag_error_elipt = zeros(1,it_max);
[coef_b_elipt,coef_a_elipt] = ellip(n,Rp,Rs,Wp_elipt);

for it=1:it_max;

    sinal = y_true+dp_ruído.*ruído(it,:);

    filt_elipt=filter(coef_b_elipt,coef_a_elipt,sinal);

    RMSE_elipt_re(it) = rmse(y_true(margem:end),filt_elipt(margem:end));
```

Anexo IV

```
lag_error_elipt(it) = rmse(sinal(1+k:end),filt_elipt(1:end-k));

end

Media_RMSE_elipt_r = mean(RMSE_elipt_re');
Media_lag_error_elipt = mean(lag_error_elipt')

DP_RMSE_elipt_r=std(RMSE_elipt_re,0,2);
DP_lag_erro_elipt=std(lag_error_elipt,0,2);

figure(5)
plot(x,sinal,'g',x,filt_elipt,'b');
title('Filtro Eliptico')
legend('Sinal com ruído','Sinal filtrado')
xlabel('Tempo')
ylabel('Sinal')

%% Filtro wavelet online %%

Jdec=2;
RMSE_wavelet_on_re = zeros(1,it_max);
lag_error_wavelet= zeros(1,it_max);

for it=1:it_max;

    sinal = y_true+dp_ruído.*ruído(it,:);

    wavelet_on = Wavelet_Online_Filter(sinal,Jdec);

    RMSE_wavelet_on_re(it) =
rmse(wavelet_on(margem:end),y_true(margem:end)');
    lag_error_wavelet(it) = rmse(sinal(2^Jdec+k:end),wavelet_on(2^Jdec:end-
k)');

end

Media_RMSE_wavelet_on_r = mean(RMSE_wavelet_on_re');
Media_lag_error_waveket_on = mean(lag_error_wavelet');

DP_RMSE_wavelet_on_r=std(RMSE_wavelet_on_re,0,2);
DP_lag_erro_wavelet=std(lag_error_wavelet,0,2);

figure(6)
plot(x,sinal,'r',x(2^Jdec:end),wavelet_on(2^Jdec:end),'k')
hold on
plot(x,y_true,'color',green)
title('Wavelet online')
legend('Sinal com ruído','Sinal filtrado')
xlabel('Tempo')
ylabel('Sinal')

%% Holt filter
alfa=;
beta=;
```

Anexo IV

```
for it=1:it_HW_max

    sinal = y_true+dp_ruído*ruído(it,:);

    [sinal_filtrado_HW,previsao_1_passo_HW] = Hot_Winters(sinal,alfa,beta,5);

    RMSE_HW_re(it) = rmse(sinal_filtrado_HW(margem:end),y_true(margem:end)');
    lag_error_HW(it) =
rmse(previsao_1_passo_HW(margem:end),sinal_filtrado_HW(margem:end));

end

Media_RMSE_HW_r = mean(RMSE_HW_re');
Media_lag_error_waveket_on = mean(lag_error_HW');

DP_RMSE_wavelet_on_r=std(RMSE_HW_re,0,2);
DP_lag_erro_wavelet=std(lag_error_HW,0,2);

figure(7)
plot(x,sinal,'r',x(2^Jdec:end),wavelet_on(2^Jdec:end),'k')
hold on
plot(x,y_true,'color',green)
title('Holt filter')
legend('Sinal com ruído','Sinal filtrado')
xlabel('Tempo')
ylabel('Sinal')

%% Resultados

% RMSE aparente medio
tab1 = [Media_lag_error_mean  Media_lag_error_exp Media_lag_error_butter...
        Media_lag_error_cheby Media_lag_error_elipt Media_lag_error_waveket_on];
disp('          RMSE aparente médio')
disp('-----')
disp('      jm_on      EWMA      Butter      Chebyshev      Eliptico      Wavelet')
disp('-----')
disp(tab1)
disp('-----')
disp(' ')
disp(' ')

% RMSE real medio
tab2 = [Media_RMSE_jm_on_r  Media_RMSE_exp_r Media_RMSE_butter_r...
        Media_RMSE_cheby_r Media_RMSE_elipt_r Media_RMSE_wavelet_on_r];
disp('          RMSE real médio')
disp('-----')
disp('      jm_on      EWMA      Butter      Chebyshev      Eliptico      Wavelet')
disp('-----')
disp(tab2)
disp('-----')
disp(' ')
disp(' ')

```

Anexo IV

```
% Desvio padrão aparente
tab3 = [DP_lag_erro_mean DP_lag_erro_exp DP_lag_erro_butter...
        DP_lag_erro_cheby DP_lag_erro_elipt DP_lag_erro_wavelet];
disp('          Desvio padrão aparente')
disp('-----')
disp('      jm_on      EWMA      Butter      Chebyshev      Eliptico      Wavelet')
disp('-----')
disp(tab3)
disp('-----')
disp(' ')
disp(' ')

% Desvio padrão real
tab4 = [DP_RMSE_jm_on_r DP_RMSE_exp_r DP_RMSE_butter_r...
        DP_RMSE_cheby_r DP_RMSE_elipt_r DP_RMSE_wavelet_on_r];
disp('          Desvio padrão real')
disp('-----')
disp('      jm_on      EWMA      Butter      Chebyshev      Eliptico      Wavelet')
disp('-----')
disp(tab4)
disp('-----')
disp(' ')
disp(' ')

% Teste t

teste_t = [RMSE_jm_on_re' RMSE_exp_re' RMSE_butter_re'...
           RMSE_cheby_re' RMSE_elipt_re' RMSE_wavelet_on_re'];

for i=1:size(teste_t,2)-1

    for ii=i:size(teste_t,2)-1
        [H(i,ii),P(i,ii)] = ttest(teste_t(:,i),teste_t(:,ii+1));
        Difer(i,ii) = mean(teste_t(:,i)-teste_t(:,ii+1));
    end

end

disp('1 se estattisticamente diferentes, 0 caso contrário')
disp(H)
format short
disp('Média das diferenças dos erros')
disp(Difer)
format shorte
disp('Valor de prova')
disp(P)

%%

%% Filtro de Kalman %%

RMSE_Kalman_re= zeros(1,it_max);
RMSE_Kalman_ap = zeros(1,it_max);

for j=1:it_max;

    sinal = y_true+dp_ruido.*ruido(j,:);
```

Anexo IV

```
data=iddata(sinal',ut',1);
[YH, FIT(nx_it), X0]=compare(data,SYS,1);

RMSE_Kalman_re(j) = rmse(y_true(margem:end)',YH.y(margem:end));
RMSE_Kalman_ap(j) = rmse(YH.y(margem:end),sinal(margem:end)');
end

Media_RMSE_kal_r = mean(RMSE_Kalman_re');
Media_RMSE_kal_ap = mean(RMSE_Kalman_ap');

DP_RMSE_kal_r=std(RMSE_Kalman_re,0,2);
DP_RMSE_kal_ap=std(RMSE_Kalman_ap,0,2);

green=[0,1.0000,0.4000];
figure(1)
plot(x,sinal,'r',x,YH.y,'k')
hold on
plot(x,y_true,'color',green)

%% Filtro de Kalman multiescala %%

Jdec=1;
dim=length(y_true);

for ii=1:it_max

sinal=y_true+dp_ruido*ruido(ii,:);
coeficients = get_coeficient(sinal,Jdec)';
coeficients_entrada=get_coeficient(entrada,Jdec)';

clear FIT
clear YH_out

for j=1:Jdec+1

data = iddata(coeficients(:,j),coeficients_entrada(:,j),1);
th = n4sid(data,'best','Ts',1);
[YH, FIT(j), X0] = compare(data,th,1);
YH_out(:,j) = YH.y;

if FIT(j)<70;
e=coeficients(:,j);
for i=1:size(coeficients(:,j),1)
YH_out(i,j)=th.C*th.K*coeficients(i,j);
end
disp('Ajuste abaixo do treshold')
disp(FIT(:,j))
end

end

wc=zeros(2^Jdec,1);
qmf=MakeONFilter('Haar',2);
L=0;
filt=zeros(dim,1);

for i=1:length(YH_out(:,1))
```

Anexo IV

```
for j=1:Jdec;
    pseud_coef=zeros(2^(Jdec-j),1);
    pseud_coef(end)=YH_out(i,j);
    wc(2^(Jdec-j):2^(Jdec-j+1)-1,1)=pseud_coef;
end
wc(1)=YH_out(i,end);

pseud_filt=IWT_PO(wc,L,qmf);
filt(i+2^Jdec-1)=pseud_filt(end);
end

RMSE_ap_MSK(ii)=rmse(filt(200:end),sinal(margem:end)'); %uncoment for test
RMSE_r_MSK(ii)=rmse(filt(200:end),y_true(margem:end)'); %uncoment for tes

end

Media_RMSE_r_MSK=mean(RMSE_r_MSK');
Media_RMSE_ap_MSK=mean(RMSE_ap_MSK');
DP_RMSE_r_MSK=std(RMSE_r_MSK,0,2);
DP_RMSE_ap_MSK=std(RMSE_ap_MSK,0,2);

figure(1)
plot(RMSE_ap_MSK)
title('RMSE_ap')
figure(2)
plot(RMSE_r_MSK)
title('RMSE_r')

figure(3)
plot(filt,'b')
hold on
plot(y_true,'g')
hold on
plot(sinal,'r')
```