

Escola Politécnica da Universidade de São Paulo

ANDRÉ AVELÃS MACHADO DE ARAUJO
RAFAEL LINO DE LIMA

**Aplicação de Filtro de Kalman ao sensoriamento de
nível baseado em sensores ultrassônicos do CH-TPN**

São Paulo

2010

ANDRÉ AVELÃS MACHADO DE ARAUJO

RAFAEL LINO DE LIMA

**Aplicação de Filtro de Kalman ao sensoriamento de
nível baseado em sensores ultrassônicos do CH-TPN**

Trabalho de formatura apresentado à Escola
Politécnica da Universidade de São Paulo.

Área de concentração: Engenharia Mecatrônica

Orientador: Prof. Dr. Eduardo Aoun Tannuri

São Paulo

2010

Ficha Catalográfica

Araujo, André Avelãs Machado de/ Lima, Rafael Lino de
Aplicação de Filtro de Kalman ao sensoriamento de
nível baseado em sensores ultrassônicos do CH-TPN/. André Avelãs
Machado de Araujo, Rafael Lino de Lima – São Paulo, 2010.

Trabalho de Formatura - Escola Politécnica da Universidade de
São Paulo. Departamento de Engenharia Mecatrônica e Sistemas
Mecânicos.

1. Filtro de Kalman 2. Ultrassom I. Universidade de São Paulo.
Escola Politécnica. Departamento de Engenharia Mecatrônica e
Sistemas Mecânicos II.t

SUMÁRIO

1. INTRODUÇÃO	5
2. METODOLOGIA	6
3. ESTUDO TEÓRICO	7
3.1. Introdução do problema	7
3.2. Apresentação do Filtro de Kalman	9
3.3. O Filtro de Kalman Discreto	10
3.3.1. Estrutura de Predição e Correção	10
3.3.1.1. Predição	10
3.3.1.2. Correção	12
3.4. Modelos Matemáticos	14
3.5. Cálculo da absorção de ondas	17
4. IMPLEMENTAÇÃO	19
4.1. Implementação para ondas regulares	20
4.1.1. Implementação <i>offline</i>	20
4.1.2. Implementação <i>online</i>	24
4.2. Implementação para ondas irregulares	25
4.2.1. Implementação <i>offline</i>	26
4.2.2. Implementação <i>online</i>	28
4.3. Problemas encontrados nas implementações <i>online</i>	28
5. AVALIAÇÃO DA ABSORÇÃO	30
5.1. Análise qualitativa	30
5.2. Análise do coeficiente de re-reflexão	32
5.3. Sensibilidade dos parâmetros do Filtro de Kalman	35
6. CONCLUSÃO	36
7. REFERÊNCIAS	37
ANEXO	38
Programa para teste preliminar <i>offline</i> de ondas regulares	38
Programa para operação <i>online</i> via Simulink de ondas regulares	40
Programa para teste preliminar <i>offline</i> de ondas irregulares	43
Programa para operação <i>online</i> via Simulink de ondas irregulares	45

1. INTRODUÇÃO

O projeto aqui apresentado propõe a utilização do Filtro de Kalman para condicionamento de sinais adquiridos por sensores ultrassônicos presentes no novo tanque de provas físico do TPN (Tanque de Provas Numérico), denominado Calibrador Hidrodinâmico (CHTPN). Estes sensores são montados na parte frontal de cada um dos 148 batedores (*flaps*) do tanque, e são responsáveis pela medida da altura da onda incidente no batedor. Esta medida é diretamente utilizada no algoritmo de absorção ativa de ondas, que é o grande diferencial deste tanque de provas.

Atualmente os sinais apresentam ruído e distorção, devido principalmente a problemas mecânicos como imperfeições na montagem dos sensores e nos dispositivos denominados “guias de onda”, que são canaletas com o intuito de melhorar as condições da superfície da água na região da medição, melhorando o sinal de ultrassom.

Visando-se obter um sinal mais preciso da altura de onda em cada *flap*, é proposta a utilização do Filtro de Kalman, que é um filtro “inteligente”, pois leva em consideração a modelagem do sistema e faz previsões da medição, ponderando essas previsões com o valor adquirido de fato no sensor. Outra característica interessante deste tipo de filtro é que ele permite a chamada fusão sensorial, ou seja, a utilização de mais de um sensor para avaliar cada medida.

Uma das primeiras aplicações notáveis deste filtro foi na estimativa de trajetória dos foguetes Apollo. Outras aplicações de destaque são para sistemas de navegação e posicionamento dos ônibus espaciais da NASA (Space Shuttle) e da Estação Espacial Internacional (ISS), além da larga utilização em sistemas de posicionamento dinâmico (DP) de navios e plataformas petrolíferas.

2. METODOLOGIA

O trabalho é dividido, basicamente, em cinco etapas principais:

- Estudo teórico
- Implementação *offline*
- Implementação *online*
- Testes e coleta de dados
- Documentação dos resultados

A etapa do estudo teórico compreende, primeiramente, a definição do problema, que deve ser feita de maneira bastante objetiva, estabelecendo claramente as condições antes do início do trabalho, bem como os resultados a que se pretende chegar. Posteriormente são estudadas tanto a formulação do Filtro de Kalman como a modelagem do problema real, anteriormente definido.

A implementação foi dividida em duas partes. A primeira, denominada *offline*, serve como avaliação da viabilidade de implantação do Filtro na prática. Nesta etapa, são utilizados sinais adquiridos de antemão em ensaios no tanque de provas, que servem de entrada para os programas de teste do Filtro. Uma vez verificados o funcionamento do Filtro e a melhoria no sinal de altura da onda, pode-se então partir para a implementação *online* no CH-TPN, que consiste, basicamente, em adaptar o programa desenvolvido na etapa *offline* de maneira que possa ser executado em tempo real, junto ao sistema de controle já existente no tanque (baseado na ferramenta Simulink do MATLAB).

Após os desafios na implementação *online*, passa-se para testes do Filtro e coleta de dados para posterior análise e avaliação final do resultado da implementação, para que se tenha uma medida objetiva da efetividade do Filtro de Kalman nesta aplicação.

3. ESTUDO TEÓRICO

3.1. Introdução do problema

O Calibrador Hidrodinâmico (CHTPN) é um tanque quadrado, com 14 m de lado e cerca de 4 m de profundidade, que opera em frequências de ondas de 0,25 até 3 Hz, e podendo gerar ondas de até 40 cm de altura [1].



Figura 3.1 – Imagem do Calibrador Hidrodinâmico do TPN

A medição da altura de onda é realizada por sensores ultrassônicos instalados em cada um dos 152 *flaps* do tanque. Aqui, vale frisar que, embora haja 152 *flaps*, apenas 148 são utilizados para geração e absorção de ondas, visto que 4 deles são fixos (para impedir a colisão dos *flaps* dos cantos do tanque). A medição é feita valendo-se da técnica de pulso-eco, ou seja, o transdutor de ultrassom emite um pulso que se propaga na direção vertical para cima, pela guia de onda, e, ao chegar à superfície da água, retorna ao transdutor. Como a velocidade de propagação do pulso de ultrassom na água é conhecida, pelo tempo entre o envio

do pulso e seu retorno (eco), é possível calcular a altura de onda média neste período. A altura de onda, bem como todo algoritmo de controle do tanque, roda a um período de amostragem de 12 ms, ou seja, a uma taxa de aproximadamente 83,33 Hz.

O problema enfrentado atualmente é que, em alguns momentos, principalmente na crista das ondas (porção superior da ondulação), existem ecos espúrios que distorcem a medição da altura de ondas, como evidenciado na figura 3.2, a seguir. Observa-se que, na subida da onda, ocorrem perdas na altura.

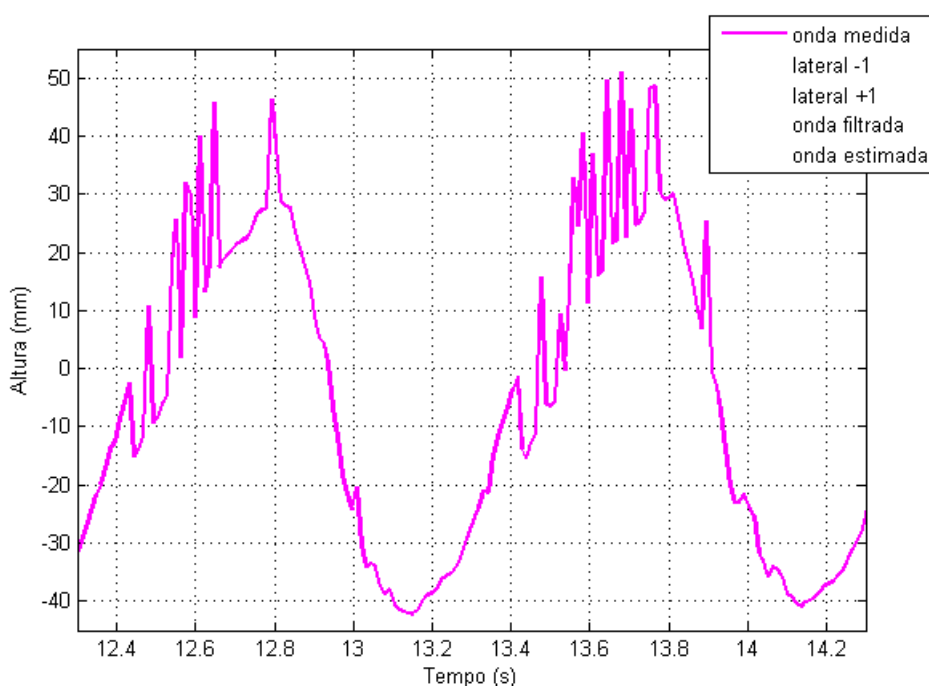


Figura 3.2 – Sinal de altura de onda em sensor ruidoso

Estes ruídos no sinal da altura de onda comprometem severamente a absorção de ondas, pois os *flaps* passam a vibrar e gerar outras ondas. A ideia, então, é utilizar o Filtro de Kalman para filtrar estas perdas de sinal e tentar obter um sinal menos ruidoso da altura de onda, e sem perdas de amplitude.

Como medida paliativa, atualmente o CHTPN utiliza dois recursos para minimizar o efeito causado pelos sinais ruidosos.

Primeiramente, é utilizando um mapeamento dos sensores e *flaps*, ou seja, *flaps* com sensores muito ruidosos utilizam os sinais de sensores dos *flaps* adjacentes (foi estabelecido um máximo de 3 *flaps* com o mesmo sensor). Isso implica que os *flaps* se movem em blocos, o que reduz a eficiência da absorção de ondas com direção diferente da perpendicular à parede.

A segunda estratégia, já para melhoria do sinal propriamente dito, é a utilização de um filtro passa baixas nos sinais de altura de onda. Este tem o inconveniente de atrasar o sinal significativamente, o que também compromete a absorção.

Vale ressaltar que, neste trabalho, há dois tipos de ondas que serão estudadas e condicionadas: ondas regulares e irregulares. As ondas regulares são ondas que possuem frequência única durante todo o ensaio. Já as ondas irregulares são geradas baseadas num espectro determinado, contendo componentes em diversas frequências.

Como metas estabelecidas no início do trabalho, acredita-se que, com a utilização do Filtro de Kalman, seja possível melhorar a absorção de ondas, que até então é de cerca de 80 a 90%, para valores entre 90 e 95%, que é o valor originalmente esperado para o CHTPN. Além disso, espera-se que o Filtro possibilite a absorção de ondas em todas as direções, isto é, deseja-se que seja possível remover o mapeamento anteriormente estabelecido e utilizar todos os sensores e *flaps* independentemente.

3.2. Apresentação do Filtro de Kalman

O Filtro de Kalman estima o estado de um processo em um determinado instante de tempo através da comparação de um modelo matemático com as medições efetuadas por sensores, ponderando entre os erros de modelagem e de medições, na busca de uma estimativa ótima. As equações utilizadas pelo Filtro podem ser divididas em dois grupos:

- equações responsáveis por projetar o estado atual, juntamente com o erro de predição, para o próximo instante de tempo, ainda sem o uso de medidas de sensores (estimativa *a priori*);
- equações responsáveis por incorporar à predição feita anteriormente uma medição realizada, obtendo-se assim uma estimativa ótima (estimativa *a posteriori*).

3.3. O Filtro de Kalman Discreto

Neste capítulo, dois conceitos fundamentais para o trabalho são abordados de forma detalhada, de acordo com as referências [2], [3] e [4]. Estes conceitos são: a estrutura de predição e correção; e o ganho do Filtro de Kalman.

3.3.1. Estrutura de Predição e Correção

Nesta seção será apresentado como um Filtro de Kalman opera e verificada sua estrutura de predição e correção (*Predictor-Corrector Structure*).

3.3.1.1. Predição

Este é o primeiro passo realizado pelo Filtro de Kalman (*Prediction Step*). Nele, o estado atual é obtido através da predição baseada no estado anterior e no modelo do sistema.

Um sistema dinâmico, modelado no Espaço de Estados, é descrito pela seguinte equação:

$$x_k = A \cdot x_{k-1} + B \cdot u_k + w_{k-1} \quad (3.1)$$

onde:

x_k é o vetor de estado; o qual se deseja estimar;

A é a matriz de estado; que relaciona o estado de um determinado passo ($k-1$) ao estado do passo atual (k). Esta matriz descreve como o estado do sistema varia entre dois instantes consecutivos e é obtida a partir do modelo físico do sistema;

B é a matriz de entrada; que relaciona o sinal de controle u_k com o estado x_k . Esta matriz também é obtida pelo modelo do sistema;

u_k é o vetor de controle;

w_k representa a incerteza do modelo. Na modelagem do Filtro de Kalman, é pressuposto que a incerteza é um sinal aleatório e tem aproximadamente uma distribuição Normal, de média nula e covariância Q :

$$p(w) \sim N(0, Q)$$

onde Q representa a incerteza do modelo.

Uma vez estimado o estado no passo k , obtém-se a amostra z_k neste mesmo passo, de acordo com o seguinte modelo:

$$z_k = H \cdot x_k + v_k \quad (3.2)$$

onde:

H é a matriz que relaciona o estado à amostra. Ela descreve como a medida depende do estado;

v_k representa o ruído de medida. Na modelagem do Filtro de Kalman, é pressuposto que o ruído de medida é aproximadamente um ruído branco Gaussiano de média nula e covariância R :

$$p(v) \sim N(0, R)$$

onde: R representa a incerteza da amostra.

Baseando-se na equação 3.1, o Filtro de Kalman determina a primeira estimativa (*a priori*) utilizando a seguinte equação de predição:

$$\hat{x}_k^- = A \cdot \hat{x}_{k-1} + B \cdot u_k \quad (3.3)$$

onde:

\hat{x}_k^- é o estado estimado *a priori* no passo k a partir do conhecimento do estado no passo $k-1$;

\hat{x}_k é o estado estimado *a posteriori* no passo k a partir do conhecimento da amostra z_k .

Ainda neste primeiro passo, o Filtro de Kalman calcula, através da seguinte equação, a incerteza na extrapolação para prever o estado seguinte (P_k^-) [3].

$$P_k^- = A \cdot P_{k-1} \cdot A^T + Q \quad (3.4)$$

3.3.1.2. Correção

No passo seguinte, o Filtro irá corrigir sua primeira estimativa, obtida no primeiro passo, baseando-se na medida recebida do sensor ultrassônico localizado no batedor. Para que essa correção seja feita, é preciso calcular o ganho de Kalman (K_k), que será apresentado na próxima seção. O resultado deste segundo passo é uma nova estimativa do estado do sistema (*a posteriori*). Pode-se verificar, na equação 3.5, abaixo, que esta nova estimativa é na verdade o estado estimado *a priori* somado de um fator de correção proporcional a diferença entre a medida adquirida e a predição de qual seria essa medida. Isto indica porque o segundo passo é conhecido como o passo de correção. É importante também ressaltar que a diferença entre a medida e a estimativa da medida é chamada Inovação ou Resíduo:

$$\hat{x}_k = \hat{x}_k^- + K_k \cdot (z_k - H \cdot \hat{x}_k^-) \quad (3.5)$$

Por fim, o erro P é atualizado de acordo com a seguinte equação [3]:

$$P_k = (I - K_k \cdot H) \cdot P_k^- \quad (3.6)$$

Monta-se, então, a estrutura do algoritmo do Filtro de Kalman da seguinte maneira:

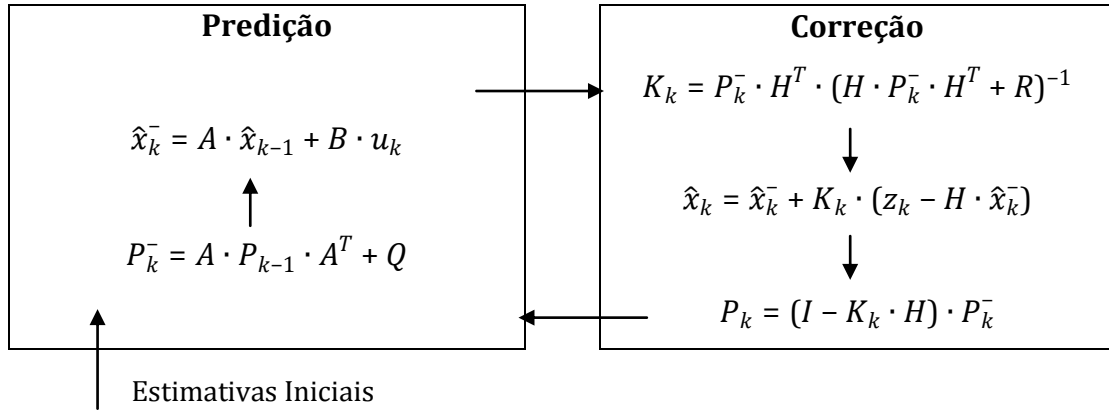


Figura 3.3 - Estrutura de Predição e Correção do Filtro de Kalman

3.3.2. Ganho do Filtro de Kalman

O ganho do Filtro de Kalman tem como objetivo minimizar a incerteza da estimativa do estado. Ele trabalha levando em consideração as ponderações que são atribuídas ao valor medido e ao valor obtido pelo modelo, ou seja, se a incerteza da medida for muito grande e o modelo bastante confiável, deve-se dar uma importância maior ao valor obtido pelo modelo, o que fará com que o ganho do Filtro de Kalman seja pequeno, e isso resulta em uma menor tendência dos resultados filtrados seguirem o sinal amostrado. Já em um segundo caso, onde o modelo gera uma grande incerteza de seus resultados, mas a medida é mais confiável, dá-se maior peso ao valor obtido pelo sensor, fazendo com que o resultado filtrado tenha uma maior tendência de seguir o sinal amostrado. Para se calcular o ganho de Kalman é utilizada a seguinte fórmula:

$$K_k = P_k^- \cdot H^T \cdot (H \cdot P_k^- \cdot H^T + R)^{-1} \quad (3.7)$$

3.4. Modelos Matemáticos

Neste capítulo, é apresentado o modelo matemático adotado, que descreve o movimento do menisco da onda que incide sobre os batedores (*flaps*), nos quais os sensores ultrassônicos se localizam.

3.4.1. Modelo de onda regular

Para pequenas amplitudes e valores de número de onda ($\kappa = 2\pi/\lambda$), como no caso do CHTPN, a equação que rege o movimento de uma onda regular na água pode ser simplificada por uma senóide. Então, a equação da altura de onda (que será medida pelos sensores ultrassônicos) é dada por:

$$x(t) = A(t) \cdot \sin(\omega t) \quad (3.8)$$

Sendo $x(t)$ a posição da superfície da água em relação ao nível zero (água parada). Para simplificar a modelagem, é utilizada a hipótese de que a amplitude da onda varia muito lentamente em relação à função seno, de maneira que sua variação pode ser desprezada. Tendo esta hipótese, para obter a equação do movimento na forma diferencial, deriva-se a equação 2.8 duas vezes, chegando a:

$$\ddot{x} = -\omega^2 x \quad (3.9)$$

Trazendo a equação 3.9 para o tempo discreto, utilizando-se os termos de primeira ordem da Série de Taylor e o método de diferenciação de diferenças regressivas, tem-se:

$$\frac{x_k - 2x_{k-1} + x_{k-2}}{\Delta t^2} = -\omega^2 x_k$$

$$\Rightarrow (1 + \omega^2 \Delta t^2) x_k = 2x_{k-1} - x_{k-2}$$

$$\therefore x_k = \frac{1}{1+w^2\Delta t^2} (2x_{k-1} - x_{k-2}) \quad (3.10)$$

A equação 3.10 mostra que o estado x_k , que se deseja estimar, é definido a partir de dois estados anteriores x_{k-1} e x_{k-2} . Por fim, escrevendo esta equação na forma matricial, resulta:

$$\begin{pmatrix} x_k \\ x_{k-1} \end{pmatrix} = \begin{bmatrix} \frac{2}{1+w^2\Delta t^2} & \frac{-1}{1+w^2\Delta t^2} \\ 1 & 0 \end{bmatrix} \begin{pmatrix} x_{k-1} \\ x_{k-2} \end{pmatrix} + \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \quad (3.11)$$

A partir desta modelagem, são obtidas as matrizes A e B da representação no Espaço de Estados:

$$A = \begin{bmatrix} \frac{2}{1+w^2\Delta t^2} & \frac{-1}{1+w^2\Delta t^2} \\ 1 & 0 \end{bmatrix} \quad \text{e} \quad B = 0.$$

É necessário agora encontrar o vetor H que relaciona o estado ao valor da medida, considerando o ruído do sinal lido pelo sensor (v_k).

$$z_k = (1 \ 0) \begin{pmatrix} x_k \\ x_{k-1} \end{pmatrix} + v_k \quad (3.12)$$

Logo,

$$H = (1 \ 0)$$

3.4.2. Modelo de onda irregular

Neste capítulo, é apresentado o modelo matemático que descreve o movimento do menisco da onda irregular. O modelo de onda utilizado no desenvolvimento deste trabalho é baseado em uma aproximação linear de segunda ordem sugerida por Sælid, Jenssen e Balchen [5], a qual pode ser escrita da seguinte forma:

$$h(s) = \frac{K_w s}{s^2 + 2 \zeta \omega_0 s + \omega_0^2} \quad (3.13)$$

onde K_w representa o ganho e é dado por:

$$K_w = 2 \zeta \omega_0 \sigma_w \quad (3.14)$$

onde ζ , por sua vez, representa o coeficiente de amortecimento; ω_0 a frequência nominal e σ_w a constante que descreve a intensidade da onda.

Esses parâmetros são dimensionados de tal forma a se obter um espectro de frequência aproximado que possua uma distribuição de energia compatível com a do espectro de frequência PM, desenvolvido por Pierson e Moskowitz em 1963 [5].

Dado o modelo que rege o movimento da onda e seus parâmetros, encontrados a partir da aproximação do espectro de frequência, pode-se escrever a equação da altura de onda na forma diferencial:

$$\ddot{y}(t) + 2 \zeta \omega_0 \dot{y}(t) + \omega_0^2 y(t) = K_w \dot{w}(t) \quad (3.15)$$

onde $y(t)$ é a posição da superfície da água em relação ao nível zero (água parada) e o $w(t)$ o ruído Gaussiano já mencionado anteriormente.

Agora, trazendo a equação 3.15 para o tempo discreto, tem-se:

$$\begin{aligned} \frac{y_k - 2 y_{k-1} + y_{k-2}}{\Delta t^2} &= -2 \zeta \omega_0 \frac{y_k - y_{k-1}}{\Delta t} + \omega_0^2 y_k \\ \therefore y_k &= \frac{2 + 2 \zeta \omega_0 \Delta t}{1 + \omega_0^2 \Delta t^2 + 2 \zeta \omega_0 \Delta t} y_{k-1} - \frac{1}{1 + \omega_0^2 \Delta t^2 + 2 \zeta \omega_0 \Delta t} y_{k-2} \end{aligned} \quad (3.16)$$

A equação 3.4 mostra que o estado y_k que se deseja estimar é definido a partir de dois estados anteriores y_{k-1} e y_{k-2} .

Por fim, escrevendo esta equação na forma matricial (Espaço de Estados), resulta:

$$\begin{pmatrix} y_k \\ y_{k-1} \end{pmatrix} = \begin{bmatrix} \frac{2+2\zeta\omega_0\Delta t}{1+\omega_0^2\Delta t^2+2\zeta\omega_0\Delta t} & \frac{-1}{1+\omega_0^2\Delta t^2+2\zeta\omega_0\Delta t} \\ 1 & 0 \end{bmatrix} \begin{pmatrix} y_{k-1} \\ y_{k-2} \end{pmatrix} + \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \quad (3.17)$$

A partir desta modelagem, são obtidas as matrizes **A** e **B** da representação no Espaço de Estados do Filtro de Kalman:

$$A = \begin{bmatrix} \frac{2+2\zeta\omega_0\Delta t}{1+\omega_0^2\Delta t^2+2\zeta\omega_0\Delta t} & \frac{-1}{1+\omega_0^2\Delta t^2+2\zeta\omega_0\Delta t} \\ 1 & 0 \end{bmatrix} \quad \text{e} \quad B = 0.$$

É necessário agora encontrar o vetor **H** do Filtro, que relaciona o estado ao valor da medida, considerando o ruído do sinal lido pelo sensor (v_k).

$$z_k = (1 \ 0) \begin{pmatrix} y_k \\ y_{k-1} \end{pmatrix} + v_k \quad (3.18)$$

Logo,

$$H = (1 \ 0)$$

3.5. Cálculo da absorção de ondas

Antes de enunciar a fórmula para cálculo da absorção, é importante entender a nomenclatura adotada por Schäffer [6] para as ondas. A figura 3.4 a seguir mostra as direções das ondas incidente, refletida e re-refletida:

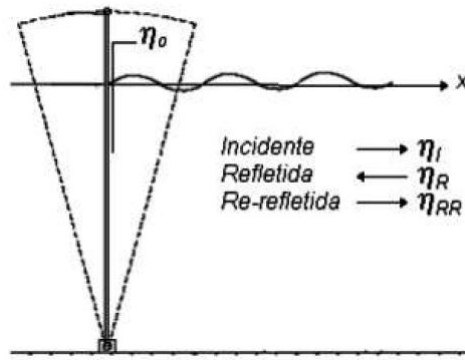


Figura 3.4 – Nomenclatura de ondas

Ou seja, é tomado como ponto de referência o modelo que está sendo ensaiado: onda incidente no modelo, refletida do modelo e re-refletida para o modelo. Para o cálculo da absorção, Schäffer [6] propõe a utilização da seguinte expressão para cálculo do coeficiente de re-reflexão:

$$RR = \frac{\tilde{A}_i - A_i}{\tilde{A}_r - A_i} \quad (3.19)$$

onde:

- A_i : amplitude da onda incidente desejada teoricamente
- \tilde{A}_i : amplitude da onda incidente obtida na prática
- \tilde{A}_r : amplitude da onda refletida obtida na prática

No caso em que há apenas absorção de ondas, $A_i = 0$. \tilde{A}_i é diferente de zero, pois, como a função de transferência do controlador é obtida numericamente, a absorção de ondas, na prática, não é perfeita (embora teoricamente seria possível absorver 100% das ondas). \tilde{A}_r é obtido pelas leituras dos sensores de altura de água no tanque.

Na figura abaixo é mostrada a função de transferência do controlador.

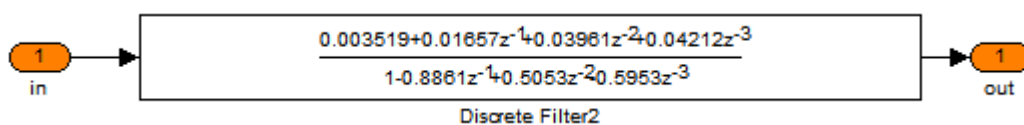


Figura 3.5 – Função de transferência do controlador de absorção, no Simulink

4. IMPLEMENTAÇÃO

Para testar o algoritmo do Filtro de Kalman, foram elaborados programas, utilizando o software MATLAB, para fazer o pós-processamento de sinais adquirido pelos sensores em outubro de 2009. A captura do sinal é feita utilizando um arquivo **.mat* que contém os dados dos ensaios realizados anteriormente.

Além da codificação padrão do Filtro, uma nova estratégia foi incluída no algoritmo, como tentativa de se obter um filtro adaptativo. Embora o Filtro de Kalman tenha sido elaborado para absorver ruídos aleatórios com distribuição probabilística normal, os sinais ruidosos do tanque de provas tendem a seguir um padrão. Desta maneira, em alguns momentos a medição é boa, e em outros a medição apresenta ruído. Assim, a ideia proposta é avaliar a Inovação (diferença entre a estimativa inicial do Filtro e o sinal medido); sendo esta diferença maior que um determinado valor, o sensor é considerado ruidoso demais, e a estimativa final é avaliada dando maior peso ao modelo que à medição. De maneira inversa, se a diferença for menor que o valor estipulado, a estimativa final tem maior influência da medição.

Os estudos preliminares do funcionamento do filtro, tanto para ondas regulares como irregulares, foram conduzidos de maneira *offline*. Uma vez verificada a viabilidade da utilização do Filtro e sua efetividade, foram desenvolvidos os programas para execução *online* do Filtro de Kalman. Nesta segunda fase, os programas implementam blocos que são incorporados à malha de controle do tanque de provas, que utiliza a ferramenta Simulink, do MATLAB, para sua operação.

4.1. Implementação para ondas regulares

4.1.1. Implementação *offline*

No ensaio analisado, os *flaps* de uma parede do tanque estão gerando ondas a 1 Hz e as outras três paredes estão fixas. A parede oposta à geradora recebe as ondas e realiza a medição das alturas de onda.

A seguir é apresentada uma plotagem de um sinal ruidoso adquirido pelo sensor de ultrassom de um *flap*, que será objeto desta análise:

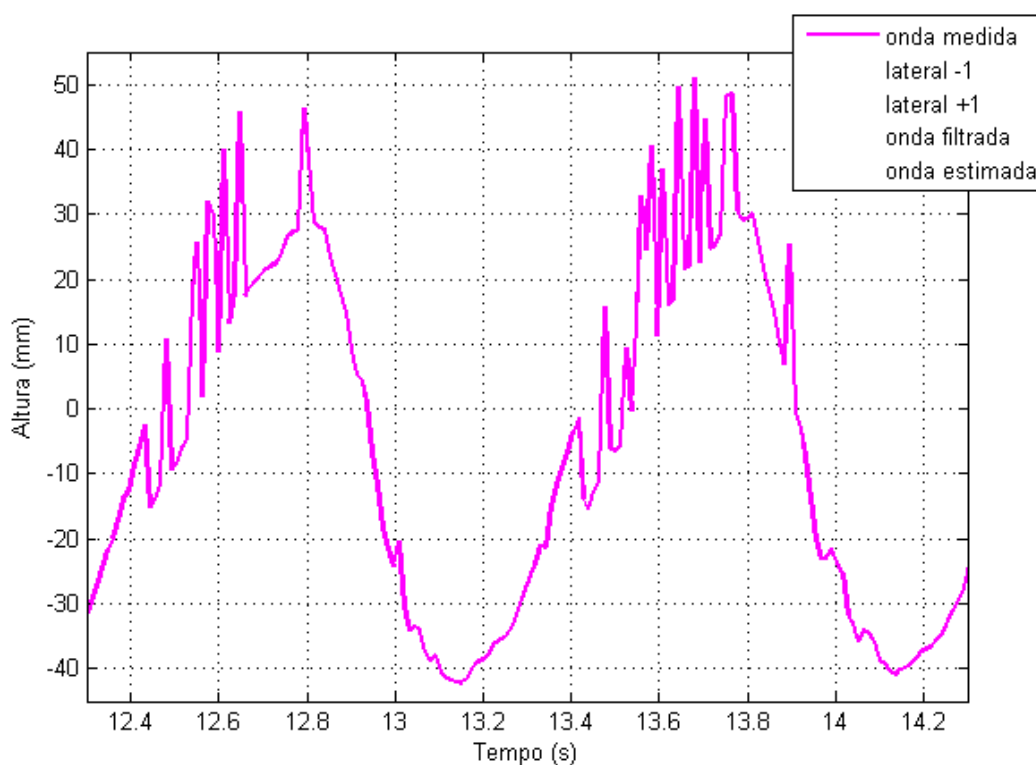


Figura 4.1 - Sinal originalmente adquirido pelo sensor de ultrassom

Percebe-se que o ruído é mais acentuado na subida da onda, e diversos pontos da medição são perdidos por conta deste problema. Para visualizar melhor o sinal perdido, pode-se observar a figura 4.2, a seguir que mostra os sinais dos sensores adjacentes ao analisado. Como a onda é perpendicular à parede que está fazendo as medições, a altura de onda é a mesma para todos os *flaps* desta parede.

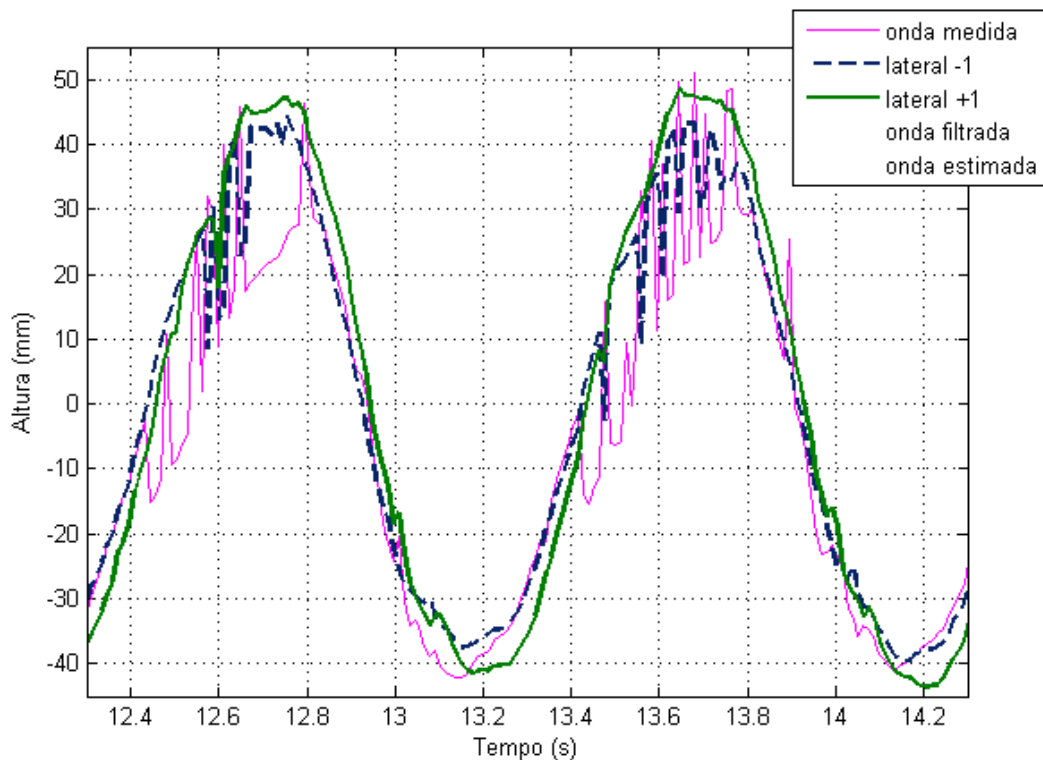


Figura 4.2 – Sinal medido do *flap* de interesse comparado com sinais dos *flaps* adjacentes

Nota-se que sensor ***lateral-1*** também apresenta medição ruidosa, porém o ***lateral+1*** possui um sinal de boa qualidade. Fica evidente a perda de amplitude do sinal medido no *flap* de interesse.

Até então, era utilizado um filtro passa baixas (do tipo Butterworth de segunda ordem, frequência de corte de aproximadamente 8 Hz) para tentar condicionar o sinal antes de sua entrada no algoritmo de absorção de ondas.

O sinal filtrado pelo filtro Butterworth é mostrado a seguir, na figura 4.3. Nesta plotagem, pode-se observar que o sinal filtrado apresenta um atraso bastante significativo em relação ao original, da ordem de 30 ms, que é aproximadamente 3 vezes o período de amostragem (12 ms). Observando também a onda medida no *flap* adjacente, nota-se a perda de informação da amplitude da onda nos picos, igualmente significativa. No caso do primeiro pico mostrado, ocorre uma perda na amplitude de até 25 mm, sendo que a onda tem amplitude de aproximadamente 80 mm.

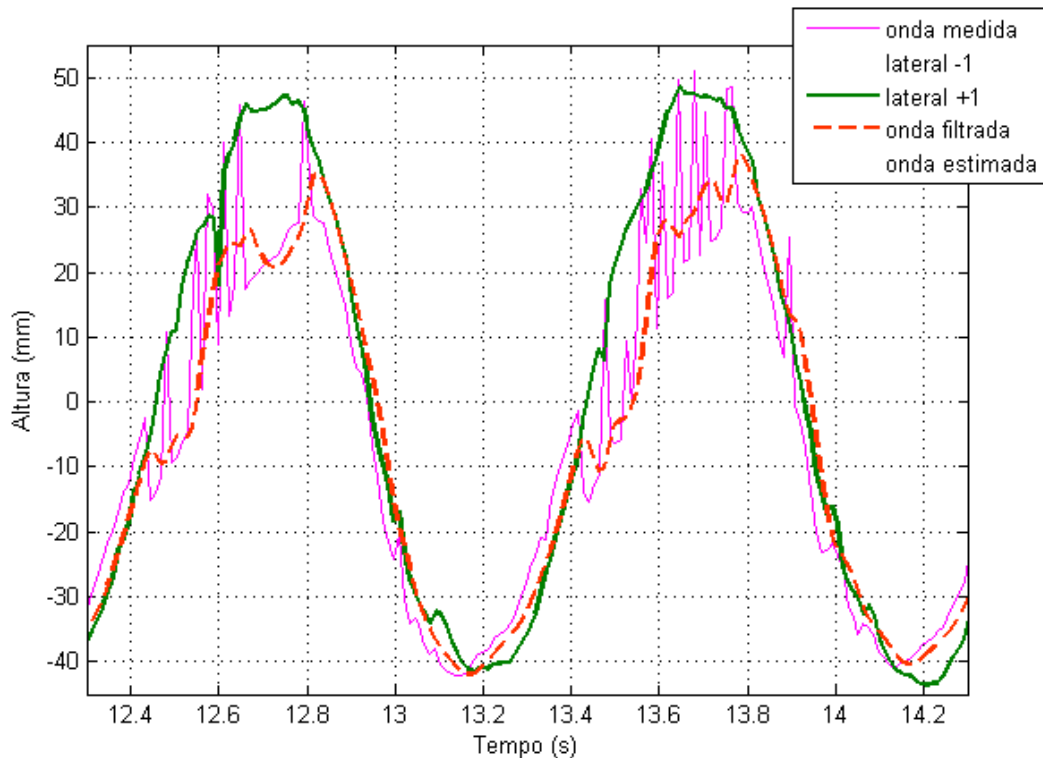


Figura 4.3 – Sinais medidos e sinal do filtro passa baixas

Com a utilização do Filtro de Kalman, já foi possível melhorar bastante o resultado da medida. Diminuiu-se o atraso entre a medição e a estimativa para valores da ordem de 10 ms, e a perda de amplitude foi diminuída no mínimo pela metade, nas cristas da onda.

Para atingir esses resultados, após a obtenção do modelo matemático e desenvolvimento do algoritmo do Filtro de Kalman, realizou-se uma bateria de testes desse algoritmo para possibilitar o ajuste dos parâmetros Q e R , que, como abordado anteriormente, representam a incerteza do modelo e ruído da medida, respectivamente.

Os valores adotados para tais parâmetros são modificados ao decorrer da estimativa, de acordo com o resultado absoluto obtido pelo cálculo da inovação, a qual é obtida pela diferença entre o valor amostrado e o valor da medida estimado pelo Filtro de Kalman. Caso a inovação esteja dentro da faixa limite, também ajustada através de testes, a medida é considerada satisfatória e então é definido o valor R_{baixa} para a covariância do erro de medida e Q_{alta} para a covariância do erro

de modelagem. Já no caso em que a inovação excede a faixa limite, assume-se R_{alta} e Q_{baixa} . O modelo de análise aqui descrito permite que o algoritmo do Filtro de Kalman apresente um caráter adaptativo.

Os valores utilizados no algoritmo desse trabalho encontram-se na tabela a seguir:

Tabela 4.1 - Parâmetros ajustados para o algoritmo do Filtro de Kalman

R_{baixa}	R_{alta}	Q_{baixa}	Q_{alta}
50	5000	5	50

As figuras a seguir mostram uma comparação entre a medição, a onda filtrada até então, e o resultado a que se pode chegar com a implementação do Filtro de Kalman:

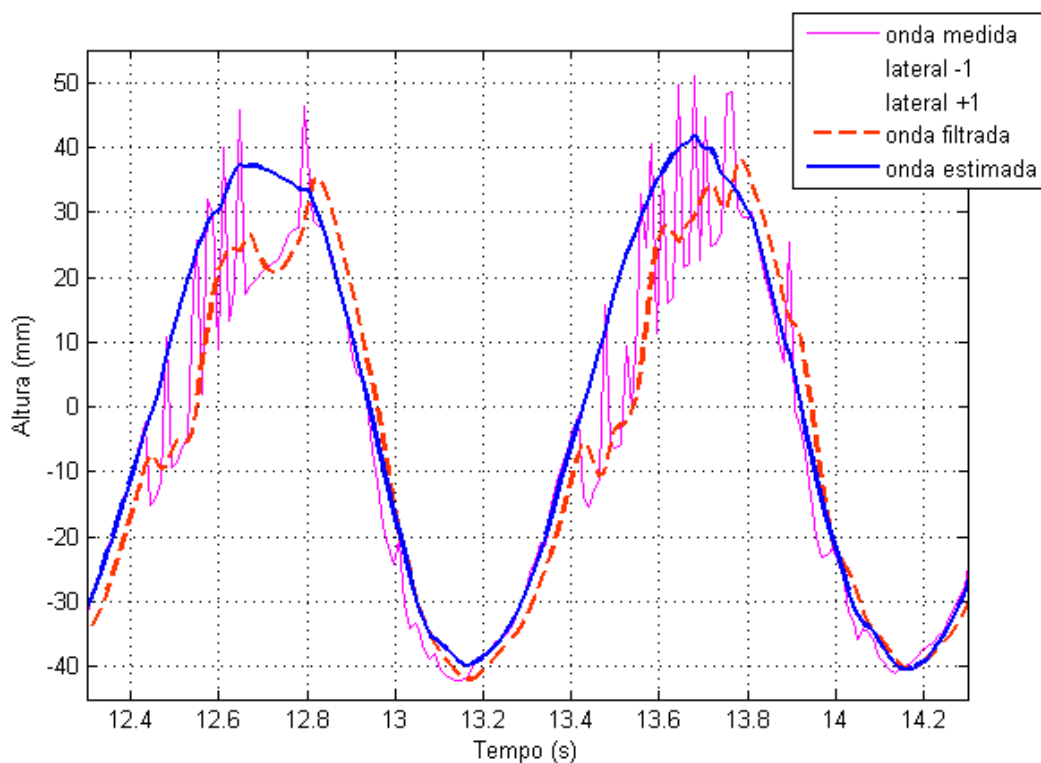


Figura 4.4 – Comparação entre sinal medido, sinal do filtro passa baixas e estimativa obtida pelo Filtro de Kalman

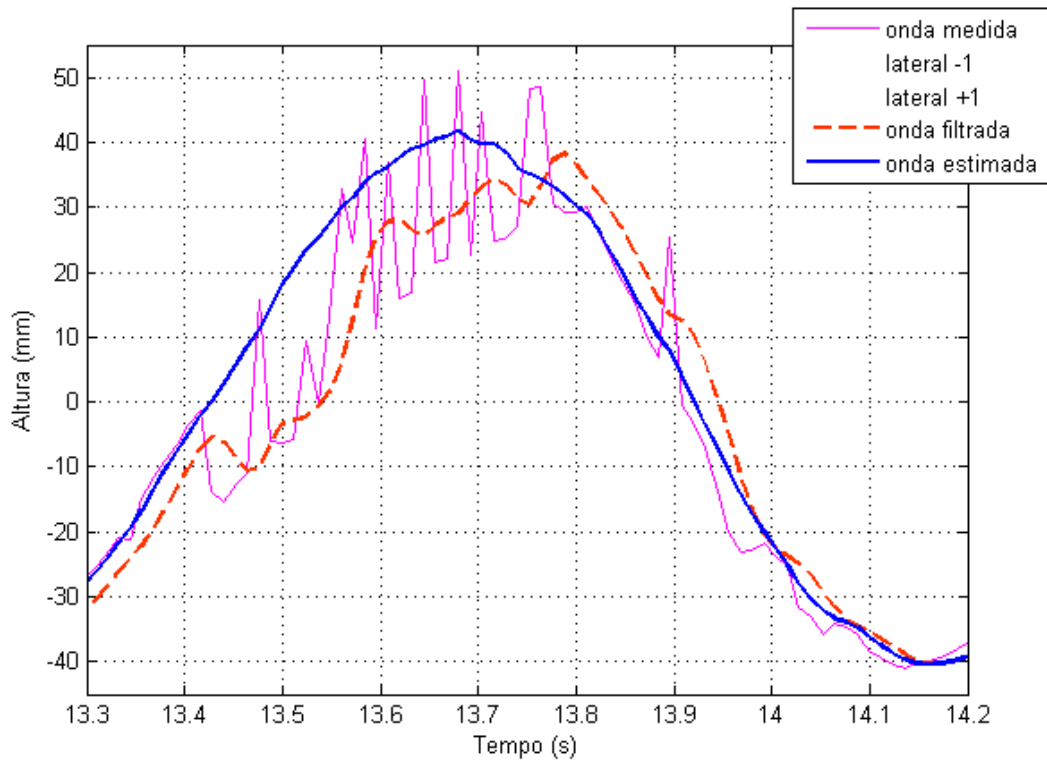


Figura 4.5 – Detalhe da comparação entre sinal medido, sinal do filtro passa baixas e estimativa obtida pelo Filtro de Kalman

4.1.2. Implementação online

A implementação *online* baseou-se no programa desenvolvido anteriormente (para a aplicação *offline*), com algumas alterações para possibilitar seu funcionamento como bloco no Simulink. A malha completa de controle do tanque de provas é mostrada na figura 4.5, abaixo:

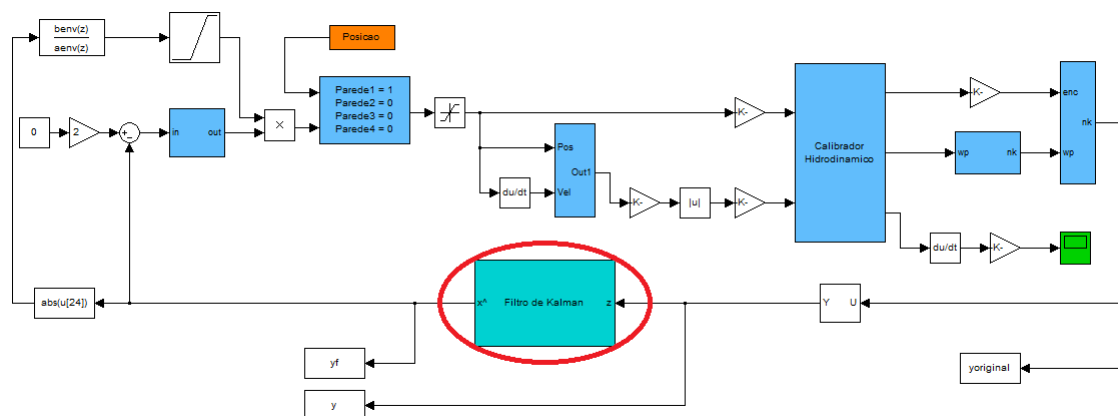


Figura 4.5 – Malha de controle do Calibrador Hidrodinâmico – Simulink

Além das alterações no software necessárias para a criação do bloco, destacado na figura anterior em vermelho, a diferença mais marcante do novo programa é que ele foi elaborado para poder fazer as operações matemáticas simultaneamente para mais de um *flap*. Isso foi realizado trabalhando-se com operações matriciais. Em vez de se utilizar um *loop* e fazer os cálculos para cada um dos *flaps* (o que tem tempo de processamento muito elevado), monta-se as matrizes do algoritmo contendo sub-matrizes correspondentes a cada um dos *flaps* utilizados, localizadas na diagonal das matrizes. As outras posições da matriz são preenchidas com zeros. Por exemplo, para a matriz de estados, **A**, o resultado fica:

$$A = \begin{bmatrix} \begin{pmatrix} \frac{2}{1+w^2\Delta t^2} & \frac{1}{1+w^2\Delta t^2} \\ 1 & 0 \end{pmatrix} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \begin{pmatrix} \frac{2}{1+w^2\Delta t^2} & \frac{1}{1+w^2\Delta t^2} \\ 1 & 0 \end{pmatrix} & & & \\ 0 & 0 & & \ddots & & \\ \vdots & & & & \ddots & \\ 0 & 0 & & & & \begin{pmatrix} \frac{2}{1+w^2\Delta t^2} & \frac{1}{1+w^2\Delta t^2} \\ 1 & 0 \end{pmatrix} \\ 0 & 0 & & & & \end{bmatrix}$$

Já a matriz **H** fica:

$$H = \begin{bmatrix} (1 \ 0) & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & (1 \ 0) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & (1 \ 0) \end{bmatrix}$$

A criação destas matrizes, bem como o algoritmo utilizado no Filtro de Kalman, podem ser verificados em mais detalhes na seção 8.

4.2. Implementação para ondas irregulares

A implementação *offline* para ondas irregulares segue os mesmo conceitos utilizados no caso das ondas regulares. O mesmo algoritmo do Filtro de Kalman, utilizado anteriormente neste trabalho, também pode ser aplicado para este caso com ondas irregulares, necessitando apenas a alteração das matrizes do modelo.

4.2.1. Implementação *offline*

A seguir, na figura 4.6, é apresentada a plotagem de uma onda irregular, tirada de uma média dos sensores de uma parede do CH-TPN.

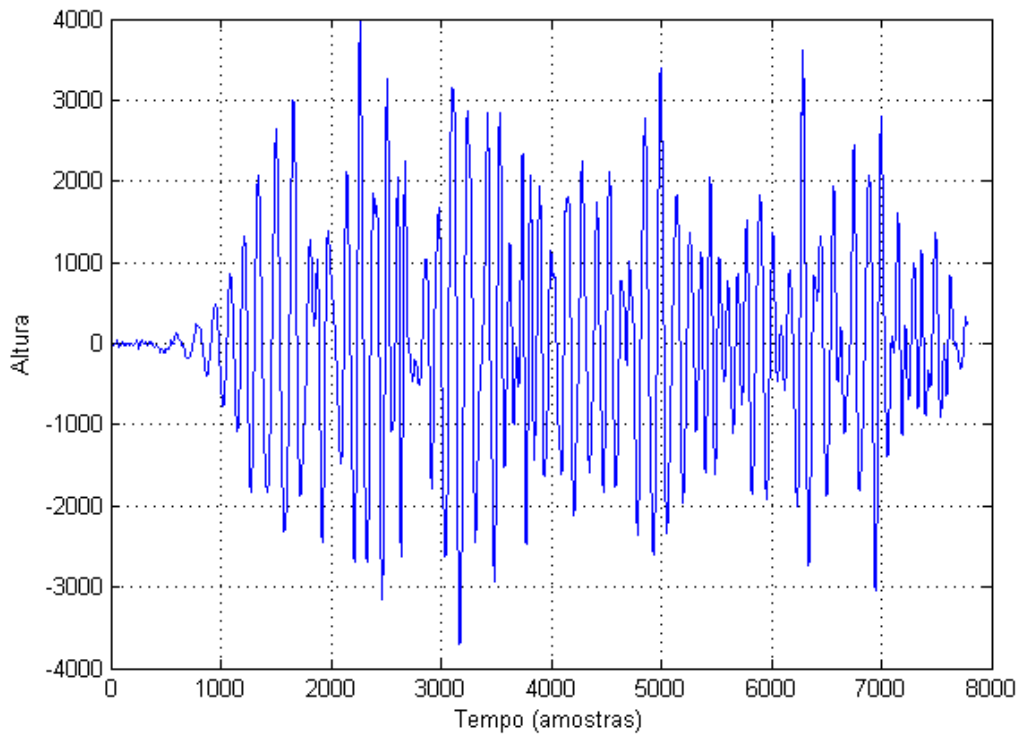


Figura 4.6 – Onda irregular avaliada pela média dos sensores de uma parede do CH-TPN

Com base neste sinal de onda, e utilizando os seguintes valores para os parâmetros do modelo descrito na seção 3.4.2, é possível gerar os espectros de frequência de onda apresentados na figura 4.7, em seguida.

$$\zeta = 0.06$$

$$\omega_0 = 0.6$$

$$\sigma_w = 4845$$

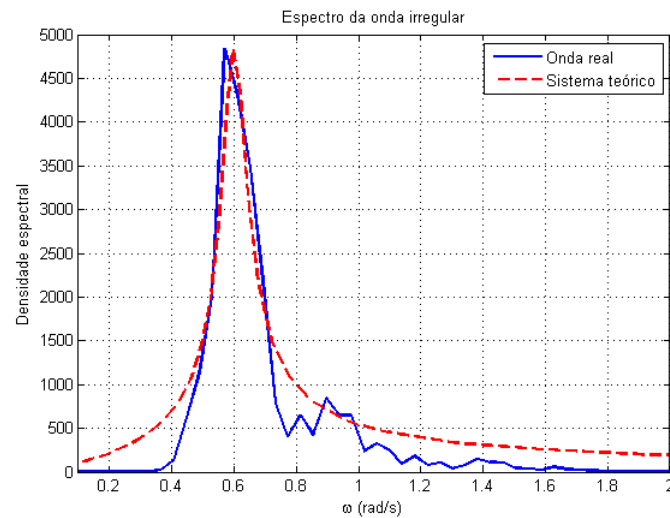


Figura 4.7 – Espectro de frequência para a onda irregular

Cabe frisar que a obtenção dos parâmetros é conduzida de forma empírica, sendo ajustados sucessivamente até que a curva seja compatível com o espectro calculado para a onda medida. Este processo busca a validação dos parâmetros calibrados a serem utilizados pelo modelo matemático. Feito isso, pode-se partir para a utilização do Filtro de Kalman.

A figura a seguir mostra uma comparação entre a medição, a onda filtrada atualmente, e o resultado que se pode chegar com a implementação do Filtro de Kalman:

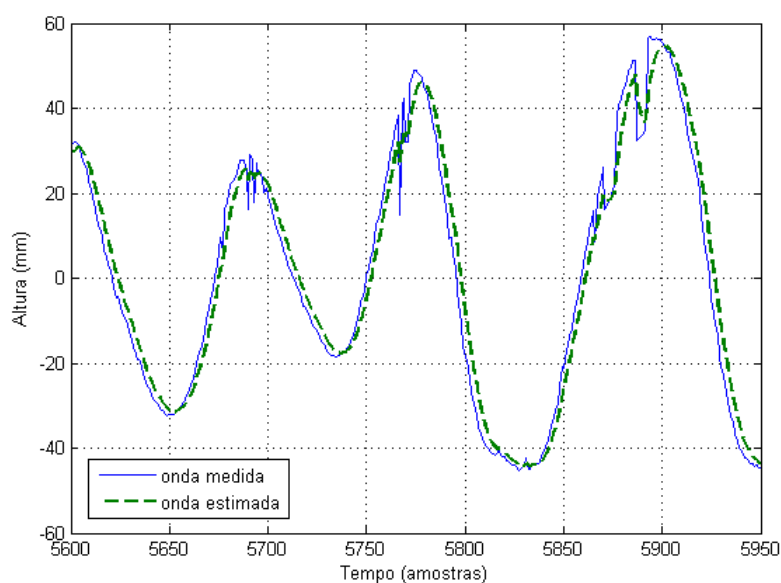


Figura 4.8 - Detalhe da comparação entre sinal medido e estimativa obtida pelo Filtro de Kalman

Assim como no caso de ondas regulares, visto anteriormente, com a utilização do Filtro de Kalman é possível melhorar o resultado da medida, reduzindo os ruídos de medida e diminuindo o atraso do sinal quando comparado a outros tipos de filtro.

Para atingir estes resultados, após a obtenção do modelo matemático e desenvolvimento do algoritmo do Filtro de Kalman, realizou-se novamente o ajuste dos parâmetros Q e R do Filtro, que, como abordado anteriormente, representam a incerteza do modelo e ruído da medida, respectivamente.

Os valores utilizados no algoritmo desse trabalho encontram-se na tabela a seguir:

Tabela 2 - Parâmetros ajustados para o algoritmo do Filtro de Kalman utilizado em ondas irregulares

R_{baixo}	R_{alto}	Q_{baixo}	Q_{alto}
500	50000	5	50

4.2.2. Implementação *online*

A implementação online do Filtro para ondas irregulares seguiu a mesma sequência adotada para ondas regulares (seção 4.1.2). Os resultados desta implementação estão apresentados no capítulo 5.

4.3. Problemas encontrados nas implementações *online*

Durante a implementação online percebeu-se que o tempo de execução do algoritmo do Filtro de Kalman é relativamente grande, e aumenta com o número de flaps utilizando o Filtro. Dependendo deste, podem ocorrer atrasos de comunicação entre o computador de controle e os CLPs.

Após a primeira versão do programa para execução de absorção *online* para ondas regulares, era somente possível trabalhar com cerca de 10 *flaps*

simultaneamente. Como o programa já estava bastante enxuto, contendo apenas os cálculos essenciais para o funcionamento do filtro, foi necessário pensar em alguma outra estratégia para atacar o problema.

Para melhorar o desempenho do programa do Filtro, foi adotada a ideia de utilizar o processador da placa gráfica do computador que controla os CLPs do tanque, que tem suporte à tecnologia CUDA (disponível nos processadores da NVIDIA [8]). Desta forma, alguns cálculos do Filtro são realizados na placa de vídeo (GPU), que tem desempenho superior ao processador do computador (CPU) para alguns tipos de cálculos. Para possibilitar a utilização do CUDA pelo MATLAB, utilizou-se o software Jacket, fornecido pela AccelerEyes [9]. Todavia, nesta topologia tem-se um tempo adicional que corresponde à transferência de dados entre a CPU e a GPU, e vice versa.

Após testes, chegou-se a uma relação que minimizou o tempo de processamento do Filtro de Kalman, enviando apenas alguns cálculos matriciais para a GPU. Com este avanço, foi possível utilizar o Filtro para uma parede completa (37 *flaps* simultaneamente).

Embora seja um avanço importante, para possibilitar a utilização do Filtro simultaneamente em todas as paredes do tanque suspeita-se que são necessárias alterações em *hardware* (investimento num computador com maior capacidade de processamento, mais rápido, já que os CLPs do tanque não podem ser trocados tão facilmente) ou utilizar o *cluster* do TPN, que possibilita paralelizar cálculos e assim reduzir o tempo de processamento. Apesar de estas opções fugirem do escopo deste trabalho, poderiam ser exploradas em trabalhos posteriores neste tema.

Outra solução que possibilitou a utilização do Filtro para uma parede completa foi a remoção do algoritmo de adaptatividade. Embora esta decisão tenha diminuído a qualidade da filtragem (em comparação com os estudos preliminares onde este recurso estava presente), a economia em tempo de processamento foi crucial. Também não foi implementada de forma *online* a utilização de mais de um sensor por batedor, pelos mesmos motivos discutidos acima.

5. AVALIAÇÃO DA ABSORÇÃO

5.1. Análise qualitativa

Nesta análise foram feitos ensaios de onda regular para 4 frequências diferentes. Uma comparação do sinal de entrada e saída do Filtro de Kalman é feita nas figuras 5.1 a 5.4, a seguir.

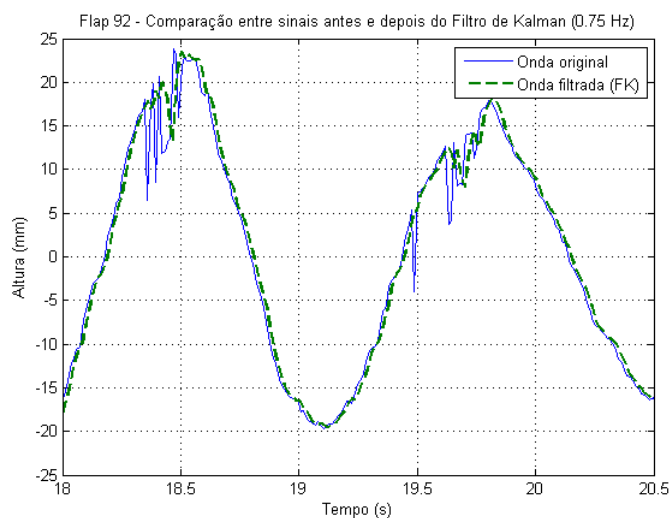


Figura 5.1 - Comparação entre sinais antes e depois do Filtro de Kalman (0,75 Hz)

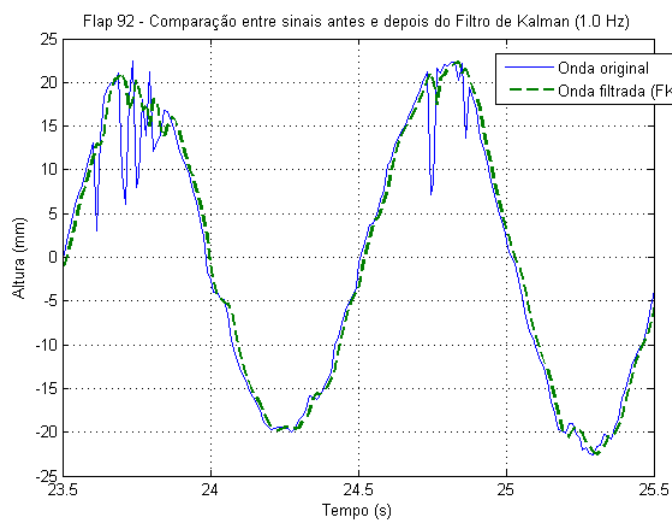


Figura 5.2 - Comparação entre sinais antes e depois do Filtro de Kalman (1,00 Hz)

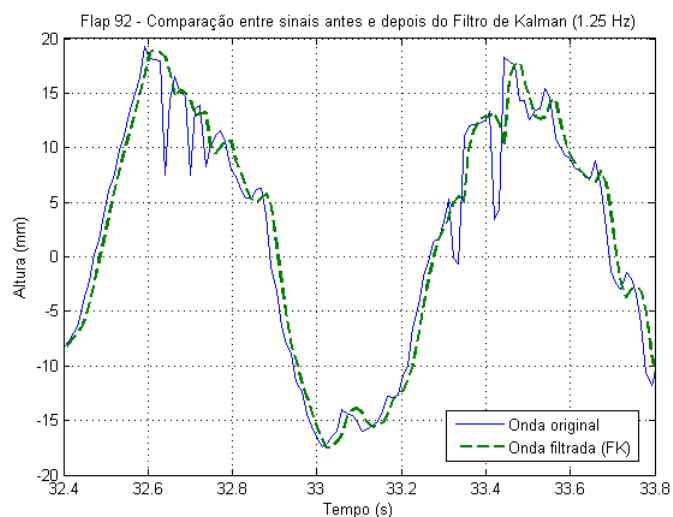


Figura 5.3 - Comparação entre sinais antes e depois do Filtro de Kalman (1,25 Hz)

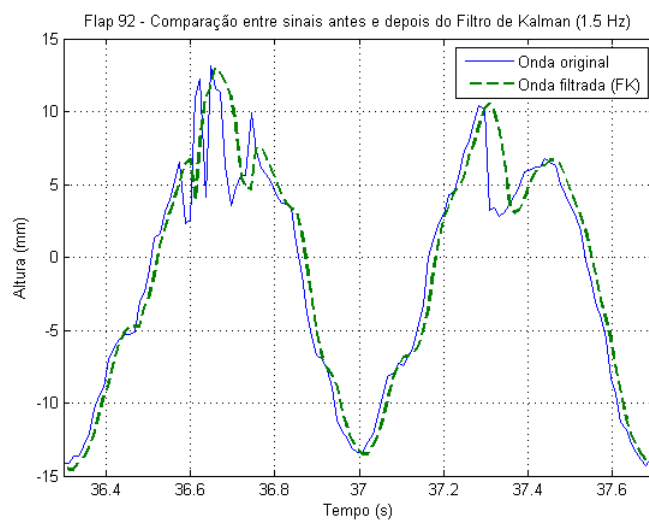


Figura 5.4 - Comparação entre sinais antes e depois do Filtro de Kalman (1,50 Hz)

Pelos gráficos, percebe-se que conforme a frequência de onda gerada aumenta, o atraso entre o sinal real e o estimado pelo Filtro também aumenta. Além disso, conforme aumenta a frequência, o ruído na altura de onda também aumenta, contribuindo ainda mais para a diminuição da qualidade do sinal.

Alem dos ensaios com ondas regulares, foi feito também um ensaio com onda irregular, que é mostrado na figura 5.5.

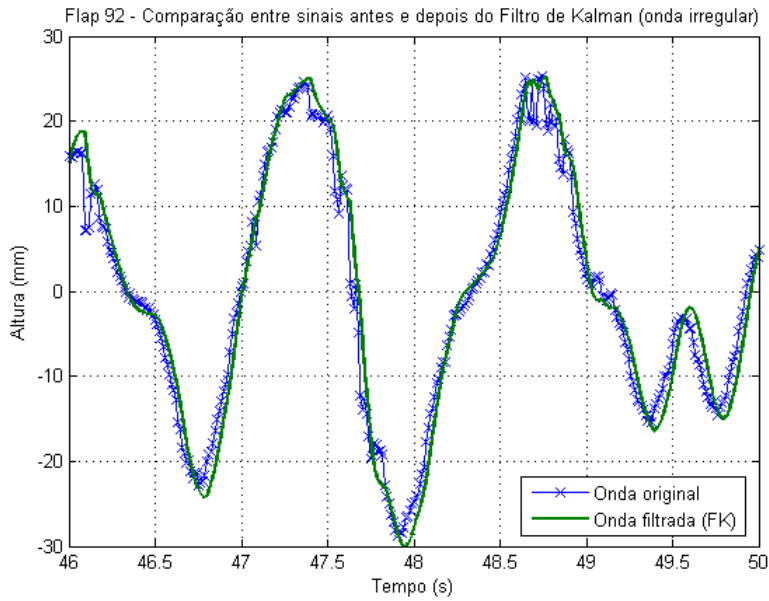


Figura 5.5 - Comparação entre sinais antes e depois do Filtro de Kalman (onda irregular)

Semelhantemente ao resultado obtido para ondas regulares, tem-se um pequeno atraso em relação ao sinal original, porém uma boa diminuição nos ruídos.

5.2. Análise do coeficiente de re-reflexão

Após a análise qualitativa, é necessário se obter em números o quanto de melhoria a utilização do Filtro de Kalman pode proporcionar para o sistema. Com o intuito de avaliar objetivamente a absorção de ondas, utilizou-se o método proposto por Schäffer [6], já discutido na seção 3.5.

Para ondas regulares, foram feitas duas análises. Em cada uma, foram realizados 4 ensaios, com ondas de frequências 0,75, 1,00, 1,25 e 1,50 Hz. A primeira análise considerou o mapeamento dos sensores até então utilizado no tanque. Na segunda, este mapeamento foi removido e todos *flaps* estavam atuando de acordo com as medições de seus respectivos sensores. Os coeficientes de re-reflexão calculados são evidenciados no gráfico a seguir. Ressaltando que o coeficiente de re-reflexão corresponde a parcela das ondas que não foram absorvidas, o desejado é que estes valores sejam tão baixos quanto possível.

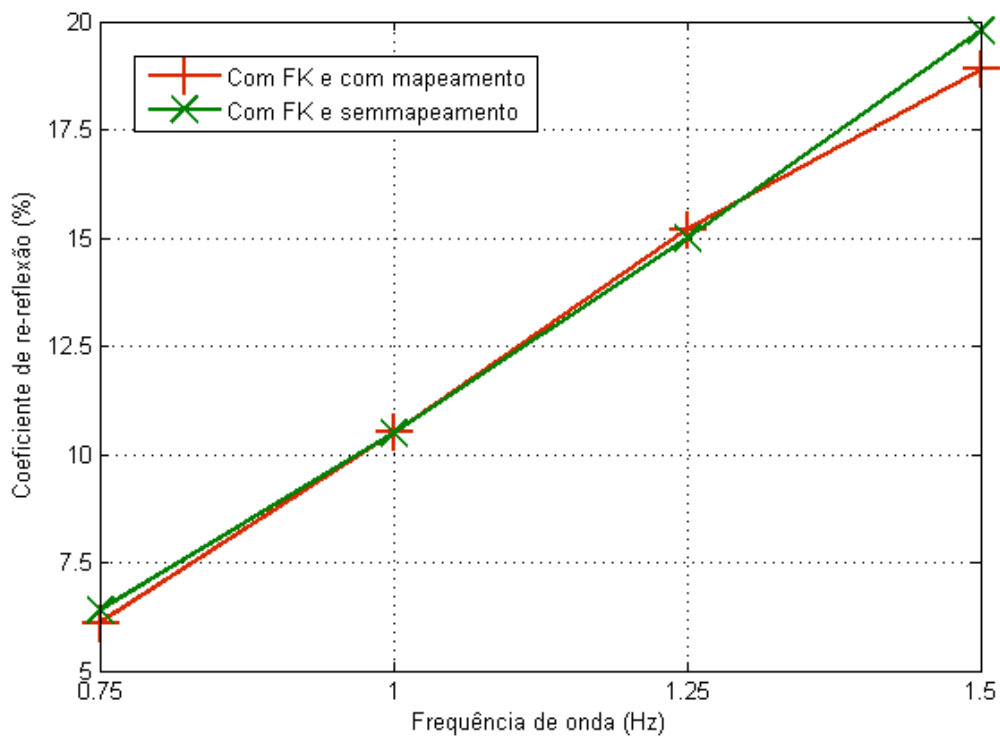


Figura 5.6 - Coeficiente de re-reflexão em função da frequência para ondas regulares

Deste gráfico resulta uma implicação muito importante para a aplicação do Filtro de Kalman neste trabalho. Como se pode observar, mesmo com a remoção do mapeamento dos sensores, ou seja, mesmo utilizando sensores mais ruidosos, a efetividade do Filtro de manteve praticamente constante. Portanto, o Filtro de Kalman pode ser utilizado para absorção de ondas com direção diferente da perpendicular à parede que está absorvendo com vantagem sobre o filtro passa baixas usado até então.

Já para ondas irregulares, foram feitos alguns ensaios com o filtro Butterworth e com o Filtro de Kalman e levantadas as curvas do coeficiente de re-reflexão em função da frequência, na região de operação do CHTPN. Os resultados são mostrados a seguir.

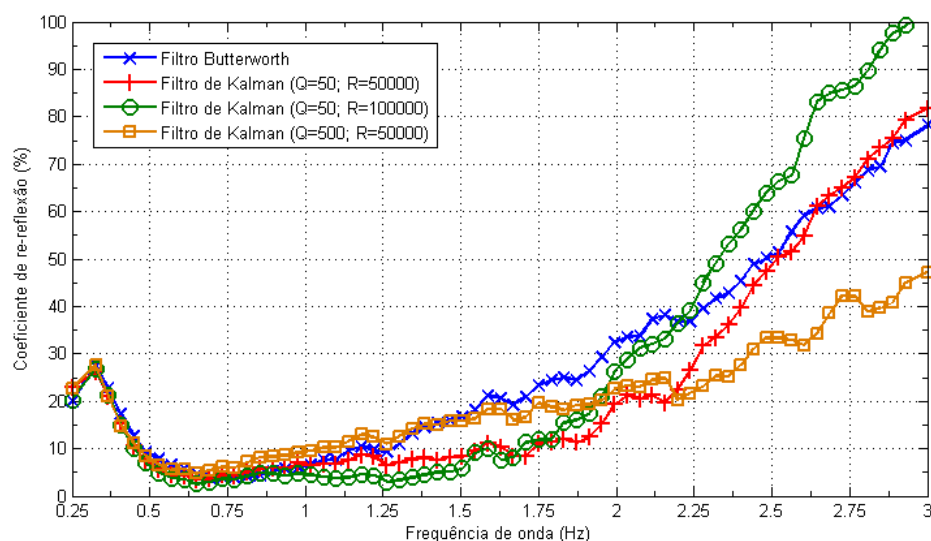


Figura 5.7 - Coeficiente de re-reflexão em função da frequência para ondas irregulares

Pode-se perceber que o Filtro de Kalman apresenta melhora significativa na absorção em algumas faixas de frequência, porém não no espectro todo. Ampliando o gráfico numa região mais utilizada nos ensaios, ficam mais evidentes os trechos em que o Filtro de Kalman supera o filtro passa baixas em desempenho.

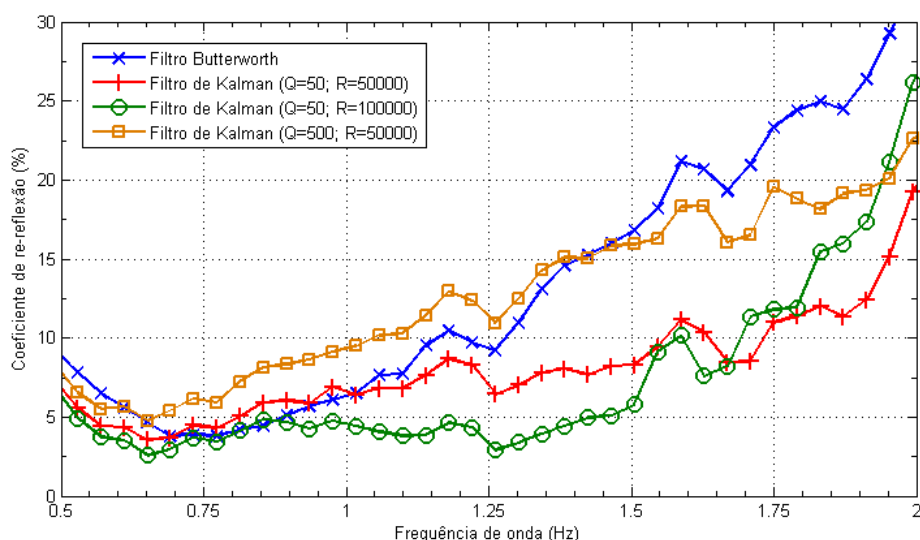


Figura 5.8 - Detalhe do coeficiente de re-reflexão em função da frequência para ondas irregulares

Constata-se então que a absorção depende da escolha cautelosa dos parâmetros do Filtro. Faz-se então necessária a realização de uma análise de

sensibilidade para a escolha correta destes parâmetros, como será visto na seção seguinte.

5.3. Sensibilidade dos parâmetros do Filtro de Kalman

Para se determinar a sensibilidade do Filtro de Kalman à variação nos parâmetros Q e R , foram calculados os coeficientes de re-reflexão, em função da frequência, para algumas combinações deles. O resultado desta análise é melhor representado no gráfico mostrado a seguir.

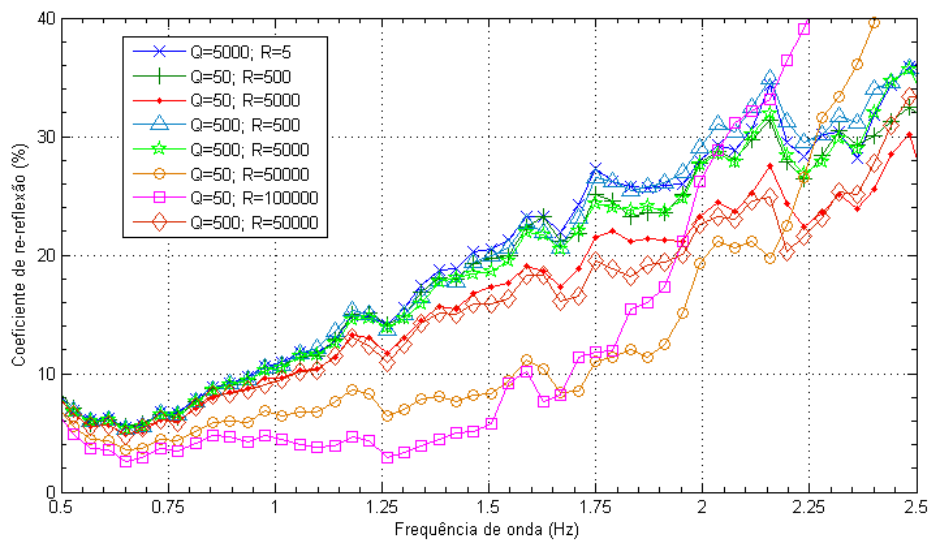


Figura 5.9 - Coeficiente de re-reflexão em função da frequência para ondas irregulares para diversas combinações de parâmetros do Filtro de Kalman

Após a visualização dos resultados, fica claro que a melhor combinação de parâmetros para o Filtro de Kalman deve ser escolhida de acordo com o espectro de ondas que será gerado. Caso este filtro seja realmente incorporado ao sistema de controle do tanque, seria possível realizar uma bateria de ensaios para se obter mais curvas do coeficiente de re-reflexão e, posteriormente, criar uma base de dados que selecione estes parâmetros automaticamente toda vez que as séries temporais das ondas são geradas no computador de controle.

6. CONCLUSÃO

A utilização do Filtro de Kalman mostrou-se uma opção bastante interessante para solucionar o problema de ruído e distorção dos sinais de altura de onda. Entretanto, ainda há desafios a serem vencidos no que diz respeito ao tempo de execução do algoritmo do Filtro dentro da malha de controle do CH-TPN.

A possibilidade de se utilizar a placa gráfica do computador de controle para realização de alguns cálculos mostrou-se uma opção interessante para diminuir o tempo de processamento do Filtro, porém suficiente apenas para funcionamento de uma parede completa do tanque.

Mesmo com as restrições de tempo de processamento apresentadas, os resultados obtidos com a utilização do Filtro de Kalman indicaram uma suavização satisfatória dos ruídos. Obteve-se ainda a redução do atraso do sinal de altura de onda, o que contribuiu consideravelmente com a melhoria, comprovada, das taxas de absorção quando comparadas com as taxas do filtro Butterworth. É importante lembrar que tal melhoria foi, desde o início, o principal objetivo deste trabalho.

A implementação do Filtro de Kalman destaca-se, também, por possibilitar a remoção do mapeamento de sensores até então existente, o que permitirá a absorção de ondas em todas as direções.

Por fim, tendo em mente que problemas como velocidade de processamento podem ser vencidos, por exemplo, com o avanço da tecnologia ou investimento em um computador mais rápido para execução do controle do tanque, o Filtro de Kalman apresenta-se como uma interessante solução capaz de melhorar as taxas de absorção do CH-TPN a níveis compatíveis com os de projeto do mesmo, independentemente de ensaios que utilizem, simultaneamente, mais de uma parede para absorção de ondas.

7. REFERÊNCIAS

- [1] TPN – **Calibrador Hidrodinâmico**. Disponível em:
< <http://www.tpn.usp.br/new/index.php/br/infra-estrutura/tanque> >
Acesso em: 25/11/2010.
- [2] KALMAN, R. E. – **A New Approach to Linear Filtering and Prediction Problems**. Transactions of the ASME – Journal of Basic Engineering, Volume 82, Series D, Pages 35-45. 1960.
- [3] CADET, O. – **Introduction to Kalman Filtering and its Use in Dynamic Positioning Systems**. Dynamic Positioning Conference. 2003.
- [4] GREWAL, M. S., ANDREWS, A. P. – **Kalman Filtering: Theory and Practice Using MATLAB**. 3rd Edition. Wiley. 2008.
- [5] FOSSEN, T. I. – **Guidance and Control of Ocean Vehicles**. Wiley. 1994.
- [6] SCHÄFFER, H. A. – **Active Wave Absorption in Flumes and 3D Basin**. Proc. Ocean Wave Measurement and Analysis. p. 1200-1208, 2001.
- [7] CARNEIRO, M. L., MELLO, P. C., LABATE, F. D., ARAUJO, A. A. M., SIMOS, A. N., TANNURI, E. A. – **USP Wave Basin: Active wave absorption and generation algorithms** – 4th International Workshop on Applied Offshore Hydrodynamics, Rio de Janeiro, Brazil. 2009.
- [8] NVIDIA – **CUDA Zone**. Disponível em:
< http://www.nvidia.com/object/cuda_home_new.html >
Acesso em: 27/11/2010
- [9] AccelerEyes – **MATLAB GPU Computing**. Disponível em:
< <http://www.accelereyes.com/> >
Acesso em: 27/11/2010

ANEXO

Programa para teste preliminar *offline* de ondas regulares

A seguir, é apresentado o programa, escrito em linguagem MATLAB, que implementa o Filtro de Kalman para uma série temporal já adquirida e armazenada em arquivo **.mat*. Com este programa é possível testar a efetividade do Filtro de maneira *offline*.

```
% Algoritmo do Filtro de Kalman - MATLAB
%
function out = kalman_onda_real(canal, f, fa, onda, onda_f)

clc;          % limpa o prompt de comando

% Declaração de variáveis

f = 1;        % frequência da onda
omega = 2*pi*f; % frequência angular da onda

fa = fa;      % frequência de amostragem
Ta = 1/fa;    % período de amostragem

t = (1:(length(onda)))*Ta; % vetor de tempo

z1 = onda(:,canal);      % sensor do flap de interesse
z2 = onda(:,canal-1);    % sensores dos flaps adjacentes
z3 = onda(:,canal+1);    %

% Declaração das matrizes para o algoritmo do Filtro de Kalman

A = [2/(1+(omega*Ta)^2) -1/(1+(omega*Ta)^2); 1 0]; % (2x2)
H = [1 0; 1 0; 1 0]; % (3x2)

Qbaixa = 5; % covariância do ruído de processo
Qalta = 50; % valores para covariância baixa e alta já pré-estabelecidos

Rbaixa = 500; % covariância do ruído de mediada
Ralta = 5000; % valores para covariância baixa e alta já pré-estabelecidos

% Atribuição inicial de valores de covariância
Q = [Qbaixa];

R = [Ralta 0 0; % matriz de covariância do ruído de mediada
     0 Ralta 0; % 3x3
     0 0 Ralta]; %

I = eye(2); % matriz identidade 2x2

% Condição inicial para execução do algoritmo:
```

```

X = [0; 0];
P = [1000];

for k = 1:length(z1)

% Predição
    X_ = A * X; % estimativa a priori
    P_ = A * P * A' + Q; % covariância do erro (2x2)

% Correção
    Z = [z1(k); z2(k); z3(k)]; % aquisição pelo sensor

    % Algoritmo de correção adaptativa das covariâncias de erro
    % (experimental)
    if k>=3
        inovacao = Z - H * X_; % cálculo da inovação (3x2)

        for i=1:3 % varredura da diagonal da matriz R
            j = 0; % variável auxiliar
            if (abs(inovacao(i)) > 5) % caso a inovação supere 5 mm
                R(i,i) = Ralta; % para cada sensor, o valor de R
                j = j+1; % é considerado alto, e o número de
            else % sensores ruidosos é incrementado
                R(i,i) = Rbaixa;
            end

            if(j>=2) % aqui é avaliado a possibilidade de 2 ou 3 sensores
                Q = Qbaixa; % fornecerem sinais ruidosos; neste caso o ruído de
            else % de processo é diminuído, dando maior peso à planta
                Q = Qalta; % na estimativa
            end
        end
    end

    K = P_ * H' / (H * P_ * H' + R); % ganho de Kalman

    X = X_ + K * (Z - H * X_); % estimativa a posteriori
    P = (I - K * H) * P_; % covariância do erro

    x1(k) = X(1,1); % armazenamento das estimativas
    K1(k) = K(1,1); % armazenamento dos ganhos de Kalman
end

% Plotagem do resultado das medições e estimativas
figure(1);
plot ( t, z1, '-m',... % plotagem da onda medida;
t, onda(:,canal-1), 'r',... % plotagem das medidas de flaps
t, onda(:,canal+1), 'g',... % adjacentes;
t, onda_f(:,canal), 'k',... % plotagem da onda filtrada;
t, x1, '-b'); % plotagem da estimativa
legend ('onda medida','lateral -1','lateral +1','onda filtrada','onda estimada');
title ('Filtro de Kalman');
xlabel ('Tempo (s)');
ylabel ('Amplitude (mm)');
grid;

% Fim do programa

```

Programa para operação *online* via Simulink de ondas regulares

Aqui, após o código de inicialização das matrizes para o algoritmo do Filtro e outras constantes pertinentes, é apresentado o programa elaborado para a incorporação do Filtro ao sistema já em utilização no Calibrador Hidrodinâmico. Este código é chamado pelo bloco do Simulink responsável pela filtragem do sinal de medição de altura.

```
% Algoritmo de inicialização
%
clear;

amp = 50;           % Amplitude de onda (em mm)
f = .75;            % Frequencia da onda (em Hz)
omega = 2 * pi * f; % Frequencia angular da onda
Ta = .012;          % Período de amostragem (em s)

nflaps = 2;         % Número de flaps a serem filtrados

% Criação da matriz A do Filtro de Kalman
for i=1:nflaps
    A( 2*i-1:2*i , 2*i-1:2*i ) = [2/(1+(omega*Ta)^2) -1/(1+(omega*Ta)^2)
                                   1 0];
end
A = sparse(A);

% Criação da matriz H do Filtro de Kalman
for i=1:nflaps
    H( i , 2*i-1:2*i ) = [1 0];
end
H = sparse(H);

% Função que cria as séries temporais para acionamento dos servo motores
[Posicao,WorkFreq,vel_us,T,b,a,benv,aenv] = ...
    regular([1.625 4.1 1.21],0.012,20,2,2,f,(amp/1000),0);

% Fim do programa
```



```

% Algoritmo do Filtro de Kalman - Simulink MATLAB
%
function FiltroKalman(block)

    setup(block);
end

function setup(block) % Função de configuração de entradas e saídas

    block.NumDialogPrms = 6;
    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;
    block.SetPreCompInPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;
    block.InputPort(1).DirectFeedthrough = false;
    block.InputPort(1).Dimensions = 148;
    block.OutputPort(1).Dimensions = 148;
    block.SampleTimes = [-1 0];
    block.SimStateCompliance = 'DefaultSimState';
    block.RegBlockMethod('PostPropagationSetup', @DoPostPropSetup);
    block.RegBlockMethod('InitializeConditions', @InitConditions);
    block.RegBlockMethod('Outputs', @Output);
end

function DoPostPropSetup(block) % Função de configuração da memória interna
                                % (para armazenar informações para a próxima iteração)
    nflaps = block.DialogPrm(1).Data;
    block.NumDworks = 2;
    block.Dwork(1).Name = 'X';
    block.Dwork(1).Dimensions = 2*nflaps;
    block.Dwork(1).DatatypeID = 0;
    block.Dwork(1).Complexity = 'Real';
    block.Dwork(1).UsedAsDiscState = true;
    block.Dwork(2).Name = 'P';
    block.Dwork(2).Dimensions = 4*nflaps^2;
    block.Dwork(2).DatatypeID = 0;
    block.Dwork(2).Complexity = 'Real';
    block.Dwork(2).UsedAsDiscState = true;
end

function InitConditions(block) % Função de inicialização

    % Variáveis globais
    global nflaps A H Qref Rref InovAlta I Q R K X_ X P P_ inovacao Z Xsaida UMSQ UMSR;

    % Inicialização
    nflaps = block.DialogPrm(1).Data;
    A = block.DialogPrm(2).Data;
    H = block.DialogPrm(3).Data;
    Qref = block.DialogPrm(4).Data;
    Rref = block.DialogPrm(5).Data;
    InovAlta = block.DialogPrm(6).Data;

    I = eye(2*nflaps); % matriz identidade (2n x 2n)
    UMSQ = ones(2*nflaps); % matriz identidade (2n x 2n)
    UMSR = ones(nflaps); % matriz identidade (2n x 2n)
    Q = eye(2*nflaps) * Qref(2); % [Qbaixa Qalta]
    R = eye(nflaps) * Rref(1); % [Rbaixa Ralta]

    block.Dwork(1).Data = zeros(1,2*nflaps); % X
    block.Dwork(2).Data = zeros(1,4*nflaps^2)+1000; % P
end

```

```

function Output(block) % Função principal do Filtro de Kalman

% Variáveis globais
global nflaps A H Qref Rref InovAlta I Q R K X_ X P P_ inovacao Z Xsaida UMSQ UMSR;

Z = block.InputPort(1).Data(1:nflaps); % o bloco adquire todos os sinais
                                         % de um determinado passo ao mesmo
                                         % tempo
X = block.Dwork(1).Data; % X = [ [x^(k-1); x^(k-2)]flap1; [x^(k-1); x^(k-2)]flap2;...];
P = reshape(block.Dwork(2).Data, 2*nflaps, 2*nflaps); % Matriz de covariância P

% Predição
X_ = A * X; % estimativa a priori
inovacao = Z - H * X_;

% Algoritmo de adaptatividade (EXPERIMENTAL)
first = find(inovacao>InovAlta)* 2 -1;
second = find(inovacao>InovAlta)* 2;
V = union(first,second);
W = second/2;
Q = (Q.*(UMSQ-sparse(V,V,1,2*nflaps,2*nflaps))) + ...
    sparse(V,V,Qref(1),2*nflaps,2*nflaps);
R = (R.*(UMSR-sparse(W,W,1,nflaps,nflaps))) + ...
    sparse(W,W,Rref(2),nflaps,nflaps);

P_ = A * P * A' + Q; % covariância do erro

% Correção
K = P_ * H' / (H * P_ * H' + R); % ganho de Kalman
X = X_ + K * (inovacao); % estimativa a posteriori
P = (I - K * H) * P_; % covariância do erro

block.Dwork(2).Data = reshape(P,1,[]);

% Saída de dados
Xsaida = reshape(X,2,[]);
Xsaida(nflaps,(nflaps+1):148) = zeros(1,(148 - nflaps));
block.OutputPort(1).Data = Xsaida(1,:);

% Armazenamento dos vetores para o próximo passo
block.Dwork(1).Data = X;
end

% Fim do programa

```

Programa para teste preliminar *offline* de ondas irregulares

A seguir, é apresentado o programa, escrito em linguagem MATLAB, que implementa o Filtro de Kalman para uma série temporal já adquirida e armazenada em arquivo **.mat* . Com este programa é possível testar a efetividade do Filtro de maneira *offline* para ondas irregulares.

```
% Algoritmo do Filtro de Kalman para onda irregular

function kalman_offline_irregular_TPN(onda, w0, zeta, t)

clc;           % limpa o prompt de comando

figure(1);
plot(t, onda);
legend('Onda medida');

flap = 31;
T = .012;

z = onda(:,flap);           % sensor do flap de interesse

% Declaração das matrizes para o algoritmo do Filtro de Kalman

A = [(2+2*zeta*w0*T)/(1 + w0^2*T^2 + 2*zeta*w0*T) -1/(1 + w0^2*T^2 + 2*zeta*w0*T)
      1 0];

H = [1 0];

Qbaixa = 5;           % covariância do ruído de processo
Qalta = 50;           % valores para covariância baixa e alta já pré-estabelecidos

Rbaixa = 500;          % covariância do ruído de mediada
Ralta = 50000;         % valores para covariância baixa e alta já pré-estabelecidos

InovAlta = 50;

I = eye(2);           % matriz identidade 2x2
Q = Qalta;
R = Rbaixa;

% Q = 5;
% R = 500;

% Condição inicial para execução do algoritmo:
X = [0; 0];
P = 1000;

for k = 1:length(z)

% Predição

X_ = A * X;           % estimativa a priori
P_ = A * P * A' + Q;   % covariância do erro (2x2)
```

```

% Correção

Z = z(k);           % aquisição pelo sensor

inov(k) = Z - H * X_;

if (k > 1)
    if (abs(inov(k)) > InovAlta)
        R = Ralta;
        Q = Qbaixa;
        disp(k)
    else
        R = Rbaixa;
        Q = Qalta;
    end
end

K = P_ * H' / (H * P_ * H' + R);   % ganho de Kalman

X = X_ + K * (inov(k));             % estimativa a posteriori
P = (I - K * H) * P_;              % covariância do erro

x1(k) = X(1,1);                    % armazenamento das estimativas
end

% Plotagem do resultado das medições e estimativas
figure(2);
plot ( t, onda(:,flap),...          % plotagem da onda medida;
      t, x1);                      % plotagem da estimativa
legend ('onda medida',...
        'onda estimada');
title (['Filtro de Kalman para onda irregular; Flap: ', num2str(flap+75)]);
xlabel ('Tempo (amostras)');
ylabel ('Amplitude');
grid;

% Fim do programa

```

Programa para operação *online* via Simulink de ondas irregulares

Aqui, após o código de inicialização das matrizes para o algoritmo do Filtro e outras constantes pertinentes, é apresentado o programa elaborado para a incorporação do Filtro ao sistema já em utilização no Calibrador Hidrodinâmico. Este código é chamado pelo bloco do Simulink responsável pela filtragem do sinal de medição de altura.

```
% Filtro de Kalman para Onda Irregular - programa de inicialização

% Carregar ensaio:
clear all;

[Posicao,WorkFreq,vel_us,T,b,a,benv,aenv]=...
    longcrested([1.625 4.1 1.21],0.012,2*60,2,2, ...
        'Jonswap1',1.66,0.092,1.611,180,5,0.5,1);

onda = Posicao.signals.values(:,1);    % carrega onda
t = length(onda);                    % vetor de tempo

% Plotagem da onda medida
figure(1);
plot(t, onda);
grid;

% Espectro da onda medida (Power, omega)
[P,w] = pwelch(onda);

% Parâmetros da função de transferência teórica do sistema
zeta = 0.05;
w0 = 0.205;
sw = 40;
Kw = 2 * zeta * w0 * sw;
h = tf([Kw 0], [1 2*zeta*w0 w0^2]);

% Cálculo do espectro do sistema
[mag fase wh] = bode(h); grid;

Ph = zeros(length(mag));
for i=1:length(mag)
    Ph(i) = mag(1,1,i);
end

% Plotagem dos espectros da onda medida e sistema teórico
figure(4);
% semilogx(w, P, wh, Ph, 'r');
plot(w, P, wh, Ph(:,1), 'r'); grid;
xlim([0.1 0.5]);
legend('Onda medida', 'Sistema teórico');
xlabel('\omega');
ylabel('PSD');

% Fim do programa
```

```

% Filtro de Kalman para Onda Irregular - bloco Simulink

function FiltroKalman_Irregular_jacket(block)
    setup(block);
end

function setup(block)
    block.NumDialogPrms = 4;

    %% Register number of input and output ports
    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;

    %% Setup functional port properties to dynamically
    block.SetPreCompInpPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;

    block.InputPort(1).DirectFeedthrough = false;
    block.InputPort(1).Dimensions = 148;

    block.OutputPort(1).Dimensions = 148;
    %% Set block sample time to inherited
    block.SampleTimes = [-1 0];

    %% Set the block simStateCompliance to default (i.e., same as a built-in block)
    block.SimStateCompliance = 'DefaultSimState';

    %% Register methods
    block.RegBlockMethod('PostPropagationSetup', @DoPostPropSetup);
    block.RegBlockMethod('InitializeConditions', @InitConditions);
    block.RegBlockMethod('Outputs', @Output);
end

function DoPostPropSetup(block)
    %% Setup Dwork
    nflaps = block.DialogPrm(1).Data(2);

    block.NumDworks = 2;

    block.Dwork(1).Name = 'X';
    block.Dwork(1).Dimensions = 2*nflaps;
    block.Dwork(1).DatatypeID = 0;
    block.Dwork(1).Complexity = 'Real';
    block.Dwork(1).UsedAsDiscState = true;

    block.Dwork(2).Name = 'P';
    block.Dwork(2).Dimensions = 4*nflaps^2;
    block.Dwork(2).DatatypeID = 0;
    block.Dwork(2).Complexity = 'Real';
    block.Dwork(2).UsedAsDiscState = true;
end

function InitConditions(block)
    global iflap nflaps A H Qref Rref InovAlta I Qst Rst UMSQ UMSR;
    %% Initialize Dwork

    iflap = block.DialogPrm(1).Data(1);
    nflaps = block.DialogPrm(1).Data(2);
    A = block.DialogPrm(2).Data;
    H = block.DialogPrm(3).Data;
    Qref = block.DialogPrm(4).Data(1:2);
    Rref = block.DialogPrm(4).Data(3:4);
    InovAlta = block.DialogPrm(4).Data(5);

```

```

I = eye(2*nflaps); % matriz identidade (2n x 2n)
UMSQ = eye(2*nflaps); % matriz identidade (2n x 2n)
UMSR = eye(nflaps); % matriz identidade (2n x 2n)
Qst = eye(2*nflaps) * Qref(2); % [Qbaixa Qalta]
Rst = eye(nflaps) * Rref(1); % [Rbaixa Ralta]

block.Dwork(1).Data = zeros(1,2*nflaps); % X
block.Dwork(2).Data = zeros(1,4*nflaps^2)+1000; % P

end

function Output(block)
    %% Filtro de Kalman
    global iflap nflaps A H Qref Rref InovAlta I Qst Rst K X_ X P P_ inovacao Z
    Xsaida UMSQ UMSR;

    Z = (block.InputPort(1).Data(iflap:(iflap + nflaps - 1))); % o bloco adquire
                                                                % todos os sinais
                                                                % de um determinado passo ao mesmo
                                                                % tempo

    % X = [ [x^(k-1); x^(k-2)]flap1 ; [x^(k-1); x^(k-2)]flap2; ... ];
    X = (block.Dwork(1).Data);

    % Matriz de covariância P (uma matriz para cada flap)
    P = (reshape(block.Dwork(2).Data, 2*nflaps, 2*nflaps));

    % Predição
    X_ = A * X; % estimativa a priori

    inovacao = (Z - H * X_);

    % Algoritmo de adaptatividade (EXPERIMENTAL)
    first = find(inovacao >InovAlta)* 2 -1;
    second = find(inovacao >InovAlta)* 2;
    V = union(first,second);
    W = second/2;

    Q = ((Qst.*(UMSQ-sparse(V,V,1,2*nflaps,2*nflaps))) + ...
        sparse(V,V,Qref(1),2*nflaps,2*nflaps));
    R = ((Rst.*(UMSR-sparse(W,W,1,nflaps,nflaps))) + ...
        sparse(W,W,Rref(2),nflaps,nflaps));

    P_ = gsingle(A * P * A' + Q); % covariância do erro

    % Correção
    K = double(P_ * H') / double(H * P_ * H' + R); % ganho de Kalman
    X = (X_ + K * (inovacao)); % estimativa a posteriori
    P = double((I - K * H) * P_); % covariância do erro

    block.Dwork(2).Data = reshape(P,1,[]);

    % Saída de dados
    Xsaida = zeros(2,148);
    Xsaida(:,iflap:(iflap + nflaps - 1)) = reshape(X,2,[]);

    block.OutputPort(1).Data = Xsaida(1,:);

    % Armazenamento dos vetores para o próximo passo
    block.Dwork(1).Data = X;
end
% Fim do programa

```