

Exercício de Programação Período 2018/02**Descrição:**

Deseja-se filtrar um sinal digital, lido de um arquivo texto, aplicando-se filtro de Kalman. O Filtro de Kalman é um filtro estocástico que é capaz de tratar sinais não estacionários. Você deve desenvolver um sistema com interface gráfica cujo layout é mostrado na Figura 1. Todo o controle da interface de ser feito usando o mouse para controlar a ação desejada.

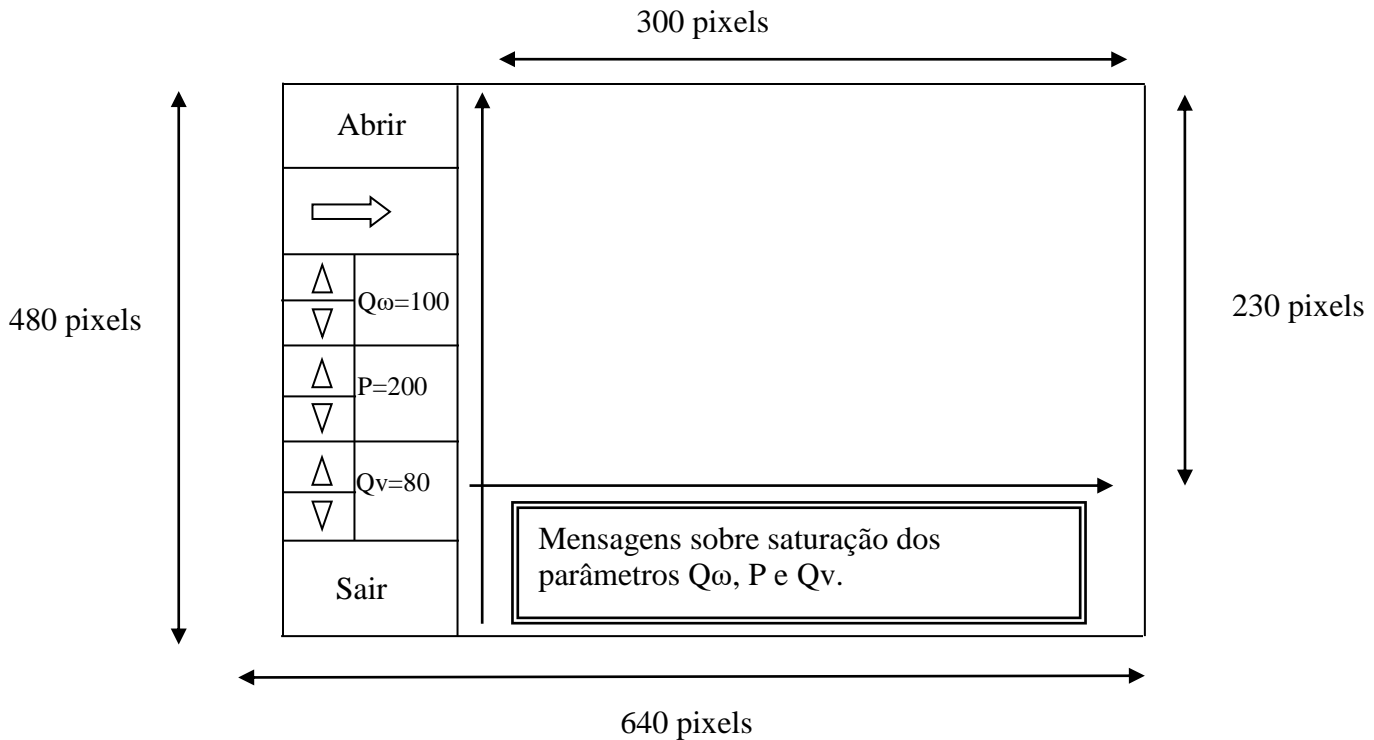


Figura 1: Layout da tela a ser desenvolvida.

Ao abrir o arquivo, contendo o sinal $x(n)$, n discreto, representado em uma *string* ASCII, deve-se convertê-lo para um vetor e, em seguida, aplicar o filtro de Kalman. Você deve plotar o gráfico do sinal coletado em azul, (com eixo horizontal conforme mostrado na Figura 1) e, sobreposto a ele, o gráfico do sinal filtrado (estimado) em vermelho. Todas as linhas da interface devem ser brancas e o fundo deve ser preto. Já para o *menu* de opções (*menu* na lateral direita), uma vez que um deles seja selecionado, deve-se trocar a cor da legenda, operação dentro de cada retângulo, de branco para verde e assim permanecer até que uma outra opção seja selecionada. A seleção deverá ser exclusivamente feita pelo uso do *mouse*. Para tanto, deve-se posicionar o cursor do *mouse* em qualquer parte do retângulo que contiver a função desejada e clicar com o botão esquerdo do mouse para executar a operação. No *menu*, tem-se as opções:

Abrir: Abrir o arquivo contendo o sinal, aqui chamado de 'sinal.txt';

→: Como a área útil do gráfico só permite 300 pontos, cada clique aqui traz mais 300 pontos; A taxa de atualização de cada ponto do gráfico deve ser de $\pm 1/18$ segs, conforme feito para o código relógio.asm.

Q_w , P e Q_v são parâmetros do filtro de Kalman. Observe que nos retângulos que contém Q_w , P e Q_v também deve ser mostrado o valor atual de cada um desses parâmetros. Por *default*, suponha que esses parâmetros sejam inicializados pelos seguintes valores: $Q_v = 10$, $P = 40$ e $Q_w = 10$. As setas para cima e para baixo incrementam/decrementam o valor do respectivo parâmetro de 20 em 20. Para todos os casos, o limite superior deve ser 500 e o limite inferior deve ser 0. Toda vez que um dos limites é atingido, deve-se colocar uma mensagem sobre a saturação em 0 ou em 500 para Q_w , P e Q_v .

O algoritmo que implementa o Filtro de Kalman pode ser observado na Figura 2. Veja que $K(n)$ é o ganho de Kalman no instante n . O que se deseja é, a partir da observação do sinal ruidoso $y(n)$, estimar $x(n)$, supondo que $x(n)$ seja descrito por um processo autoregressivo de ordem 1, AR(1), conforme as equações a seguir

$$x(n+1)=A(n)x(n)+B(n)\omega(n) \quad , \quad (1)$$

$$y(n)=C(n)x(n)+v(n) \quad . \quad (2)$$

x é a variável de estado, representado por um modelo linear com parâmetros variam no tempo. $y(n)$ representa o modelo do sensor, ou seja, deseja-se medir x mas observa-se y . Diz-se que x é uma variável latente (no sentido de escondida). $\omega(n)$ e $v(n)$ são caracterizados por ruído branco gaussiano, com potência (variância) $Q\omega$ e pot_cont , respectivamente. Nesse trabalho, deve-se fixar $pot_cont=1$. P é a potência (variância) do erro entre x_{prox} e x_{ant} . P é usado no algoritmo como p_{ant} e p_{prox} . Q_v é a potência do ruído do sensor (cuja variável aleatória é $v(n)$). Apesar do algoritmo apresentado na Figura 2 ser matricial, aqui, assume-se para essa implementação de um sinal 1D, $A=1$, $B=1$, $C=1$ para todo n . Algoritmo disponível em M. Hayes, "Statistical Digital Signal Processing and Modeling", Wiley, 1996, pp 377. Na Figura 2, o super-índice H diz respeito ao operador hermitiano de uma matriz (transposta conjugada de uma matriz). Como a implementação é 1D e os parâmetros e sinais são todos reais, a aplicação do hermitiano a um escalar resulta no próprio escalar.

Table 7.4 The Discrete Kalman Filter

State Equation	$\mathbf{x}(n) = \mathbf{A}(n-1)\mathbf{x}(n-1) + \mathbf{w}(n)$
Observation Equation	$\mathbf{y}(n) = \mathbf{C}(n)\mathbf{x}(n) + \mathbf{v}(n)$
Initialization:	$\hat{\mathbf{x}}(0 0) = E\{\mathbf{x}(0)\}$ $\mathbf{P}(0 0) = E\{\mathbf{x}(0)\mathbf{x}^H(0)\}$
Computation:	For $n = 1, 2, \dots$ compute $\hat{\mathbf{x}}(n n-1) = \mathbf{A}(n-1)\hat{\mathbf{x}}(n-1 n-1)$ $\mathbf{P}(n n-1) = \mathbf{A}(n-1)\mathbf{P}(n-1 n-1)\mathbf{A}^H(n-1) + \mathbf{Q}_w(n)$ $\mathbf{K}(n) = \mathbf{P}(n n-1)\mathbf{C}^H(n) \left[\mathbf{C}(n)\mathbf{P}(n n-1)\mathbf{C}^H(n) + \mathbf{Q}_v(n) \right]^{-1}$ $\hat{\mathbf{x}}(n n) = \hat{\mathbf{x}}(n n-1) + \mathbf{K}(n) \left[\mathbf{y}(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n n-1) \right]$ $\mathbf{P}(n n) = \left[\mathbf{I} - \mathbf{K}(n)\mathbf{C}(n) \right] \mathbf{P}(n n-1)$

Figura 2: Algoritmo do filtro de Kalman, retirado de M. Hayes, "Statistical Digital Signal Processing and Modeling", Wiley, 1996, pp 377.

A qualquer instante você pode aplicar qualquer uma das funções do menu lateral. Ao clicar em ‘Abrir’ com um arquivo já aberto, seu código deve fechar o arquivo aberto para, novamente, abri-lo. As cores na tela são de acordo com a Tabela 1.

Tabela 1: Tabela de Cores modo VGA 640×480

Preto	0	Cinza	8
Azul	1	Azul_claro	9
Verde	2	Verde_claro	10
Cyan	3	Cyan_claro	11
Vermelho	4	Rosa	12
Magenta	5	Magenta_claro	13
Marrom	6	Amarelo	14
Branco	7	Branco_intenso	15

Para facilitar a programação, um programa (**LINEC.ASM**) é fornecido, contendo o básico para a mudança do modo de vídeo (gráfico, 640×480 com 16 cores), funções de plotar ponto, desenhar uma linha, posicionar o cursor e escrever um caractere. Este programa deve ser usado como referência para a programação do exercício. Para as funções gráficas, a tela pode ser vista como tendo 640×480 (x e y) sendo que a origem (0,0) é o ponto inferior à esquerda da tela.

Procedimento:

O programa deverá abrir o arquivo e ler o seu conteúdo para uma variável na memória e depois fechar o arquivo. A partir desta variável os dados devem ser processados para mostrar as figuras descritas no arquivo. Utilizar as seguintes funções da INT 21H :08H,3DH,3EH,3FH, e 4CH e as interrupções da BIOS das funções de vídeo (INT 10H). Para a utilização do mouse, utilize a INT33H, cuja forma de utilização é semelhante à INT21H (veja uma descrição detalhada em http://stanislavs.org/helppc/int_33.html). O arquivo a ser lido, sinal.txt, pode, portanto, ser aberto no Matlab, tem cada amostra codificada em uma string de ASCII. Por exemplo:

a string de ASCII 2.9000000e+001 representa 29 ,

a string de ASCII 7.0000000e+000 representa 7 ,

Cada amostra deve variar na faixa inteira de 0 a 255. Seu programa deve abrir o arquivo, converter cada amostra de string no formato ASCII para um valor inteiro de 8 bits e com sinal. Para plotar o sinal, você precisa fazer no Matlab (você vai carregar o arquivo sinal.mat no matlab):

```
>> load -ascii sinal;
```

```
>> plot(sinal);
```

Se quiser testar com outros sinais, por exemplo, um sinal gravado por você, escolha a taxa de amostragem e use ou os recursos do S.O. ou comandos do próprio Matlab para fazê-lo. Só tenha cuidado com a duração e as conversões. O teste de avaliação que irei fazer será com o arquivo sinal.txt .

Importante: Procedimento para entrega:

Enviar para o email: evandro.salles@ufes.br com o assunto: “exe_sistemb1_2018-2”. O nome do arquivo ASM deve ser as iniciais de seu nome e deve conter, no máximo, 8 caracteres. No arquivo, não se esqueça de adicionar logo nas primeiras linhas, comentadas, o seu nome completo e a turma. Enviar até 22/10/2018, às 23:59:59 h.

Atenção:

O exercício é individual. Qualquer cópia parcial ou total acarretará na atribuição da nota 0 (zero) para todos os envolvidos.