UNIVERSIDADE ESTADUAL DO MARANHÃO - UEMA

CENTRO DE CIENCIAS TECNOLOGICAS - CCT

PÓS GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO E SISTEMAS- PECS

FILTRAGEM ÓTIMA COM APLICAÇÕES AEROESPACIAIS

JOSE RIBAMAR RIBEIRO SILVA JUNIOR

JACYEUDE DE MORAIS PASSOS ARAÚJO SEGUNDO

MOISÉS JOSÉ DOS SANTOS FREITAS

**Relatório 3 – Filtro de Kalman Discreto**

São Luís

2017

**Método de *Monte Carlo***

O método de Monte Carlo é um método numérico universal para resolver problemas por meio de amostragem aleatória, este se relaciona com distribuição aleatória de estados em espaços livres em um mapa.

A única exigência é que seja feita a modelagem em termos de funções de densidade de distribuição de probabilidade (FDP). Após isso, a simulação de Monte Carlo pode fazer amostragens aleatórias a partir destas equações. O processo é repetido várias vezes e o resultado é obtido por meio de estatísticas que utilizam média e desvio padrão por exemplo, sobre um número de amostras que pode chegar a milhões.

**Filtro de Kalman**

O filtro de Kalman é um conjunto de equações matemáticas que realiza um processo de estimação com erro quadrático minimizado. Realiza-se a observação da variável de observação para que a variável de estado seja estimada eficientemente (estimando os estados passados, do presente e do futuro). É aplicado para modelos escritos na forma de espaço de estado e permite a estimação dos parâmetros desconhecidos através da maximização da verossimilhança com a decomposição de erro de previsão.A estimação pode ser feita utilizando estimação em batelada e também estimação recursiva.

**IMPLEMENTAÇÃO DO CÓDIGO NO MATLAB**

%Modelo

%Matrizes A,B,C

A = [1 0.1;0 1];

B = [0.005;0.1];

C = [1 0];

%Matrizes R e Q para a estimação dos parâmetros

Q = [0.01 0;0 0.04];

R = 0.01;

%Valores de P e x para o cálculo da precisão dos modelos

P\_ = [1e-4 0;0 1e-8];

x\_ = [1;0];

yk\_0 = 5;

Ts = 0.1;

T = 30;

Nr = 1;

e1 = [1;0];

e2 = [0;1];

x1k = zeros(2,T/Ts); %Estado verdadeiro do sistema

y1k = zeros(1,T/Ts); %Saída verdadeira do sistema

xe1kk = zeros(2,T/Ts); %Estado estimado pelo Filtro de Kalman

ye1kk = zeros(1,T/Ts); %Saída estimada pelo Filtro de Kalman

P1kk = zeros(2,2,T/Ts); %Variância

Pkky = zeros(1,T/Ts); %Variância da saída

Pkkxy = zeros(2,T/Ts); %Variância cruzada

K1k = zeros(2,T/Ts); %Ganho de Kalman

erro = zeros(2,T/Ts); %Erros de X1 e X2

sigma1 = zeros(1,T/Ts); % Desvio padrão de X1

sigma2 = zeros(1,T/Ts); % Desvio padrão de X2

for j = 1:Nr %Monte Carlo

%Simulação

x1k(:,1) = x\_ + sqrt(P\_)\*randn(2,1);

vk = sqrt(R)\*randn;

y1k(1) = C\*x1k(:,1) + vk;

%Filtro de Kalman - Inicialização

xe1kk(:,1) = x\_;

P1kk(:,:,1) = P\_;

ye1kk(:,1) = C\*xe1kk(:,1);

P1kk(:,:,1) = A\*P1kk(:,:,1)\*A'+Q;

Pkky(:,1) = C\*P1kk(:,:,1)\*C'+R;

Pkkxy(:,1) = P1kk(:,:,1)\*C';

K1k(:,1) = Pkkxy(:,1)\*inv(Pkky(:,1));

xe1kk(:,1) = xe1kk(:,1) + K1k(:,1)\*(y1k(1)-ye1kk(:,1));

P1kk(:,:,1) = P1kk(:,:,1) - Pkkxy(:,1)\*inv(Pkky(:,1))\*Pkkxy(:,1)';

%Cálculo do erro

erro(1,1) = e1'\*(x1k(:,1)-xe1kk(:,1)); %erro de X1

erro(2,1) = e2'\*(x1k(:,1)-xe1kk(:,1)); %erro de X2

%Desvio padrão

sigma1(1,1) = sqrt(P1kk(1,1,1));

sigma2(1,1) = sqrt(P1kk(2,2,1));

for i = 2:T/Ts

%Calculo do x1k

wk = sqrt(Q)\*randn(2,1);

uk = 10\*(yk\_0 - e1'\*x1k(:,i-1)) - 2\*e2'\*x1k(:,i-1);

x1k(:,i) = A\*x1k(:,i-1) + B\*uk + wk;

%Calculo do y1k

vk = sqrt(R)\*randn;

y1k(i) = C\*x1k(:,i) + vk;

%Filtro de Kalman - Predição

xe1kk(:,i) = A\*xe1kk(:,i-1) + B\*uk;

ye1kk(:,i) = C\*xe1kk(:,i);

P1kk(:,:,i) = A\*P1kk(:,:,i-1)\*A'+Q;

Pkky(:,i) = C\*P1kk(:,:,i)\*C'+R;

Pkkxy(:,i) = P1kk(:,:,i)\*C';

%Filtro de Kalman - Atualização

K1k (:,i) = Pkkxy(:,i)\*inv(Pkky(:,i));

xe1kk(:,i) = xe1kk(:,i) + K1k(:,i)\*(y1k(i)-ye1kk(:,i));

P1kk(:,:,i) = P1kk(:,:,i) - Pkkxy(:,i)\*inv(Pkky(:,i))\*Pkkxy(:,i)';

%Cálculo do erro

erro(1,i) = e1'\*(x1k(:,i)-xe1kk(:,i)); %erro de X1

erro(2,i) = e2'\*(x1k(:,i)-xe1kk(:,i)); %erro de X2

%Desvio padrão

sigma1(1,i) = sqrt(P1kk(1,1,i));

sigma2(1,i) = sqrt(P1kk(2,2,i));

end

yk\_ = zeros(1,T/Ts);

for i = 1:T/Ts

yk\_ (1,i) = 5;

end

%Posição vertical

figure(1);

hold on;

plot(y1k);

plot(ye1kk);

plot(yk\_);

title('Posição vertical');

%X1

figure(2);

hold on;

plot(xe1kk(1,:));

plot(x1k(1,:));

title('x1');

%X2

figure(3);

hold on;

plot(xe1kk(2,:));

plot(x1k(2,:));

title('x2');

%Erro de X1

figure(4);

hold on;

plot(erro(1,:));

plot(sigma1(1,:));

plot(-sigma1(1,:));

title('Erro de x1');

%Erro de X2

figure(5);

hold on;

plot(erro(2,:));

plot(sigma2(1,:));

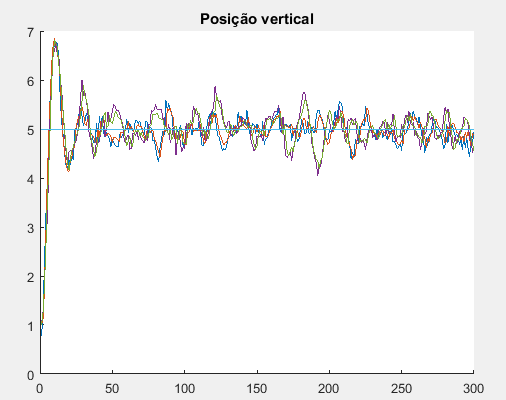
plot(-sigma2(1,:));

title('Erro de x2');

end

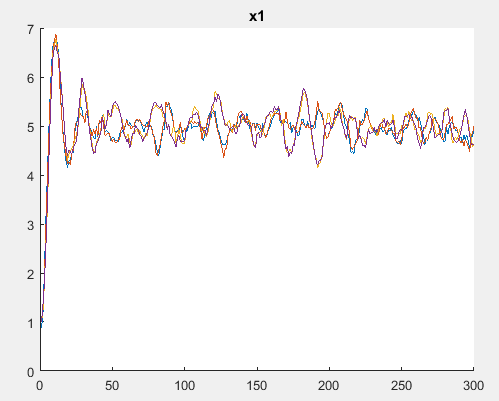
**RESULTADOS**

De acordo com a figura 1, nota-se que os valores de saída de tendem a estabilizar partir do ponto de aproximadamente 40. Há um salto da curva no gráfico das respostas das estimativas e depois as estimativas tendem ao regime permanente. A curva das estimativas tendem a acompanhar a referência até o fim do processo, indicando uma eficiência do processo.

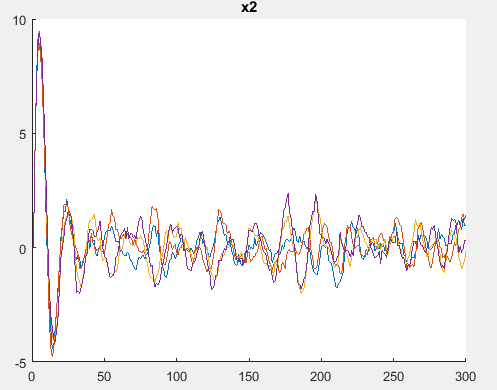
****

**Figura 1**

As figuras 2 e 3 mostram as estimativas de valores reais de x1 e x2 em relação aos valores estimados pelo Filtro de Kalman. Nota-se uma aproximação entre as curvas dos valores estimados e dos valores reais de x, tendendo a uma estabilização após um overshoot.

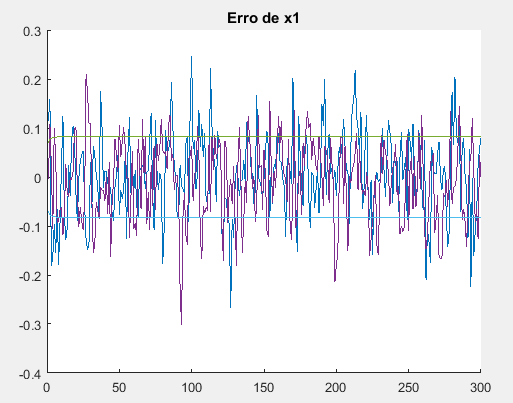
****

**Figura 2**

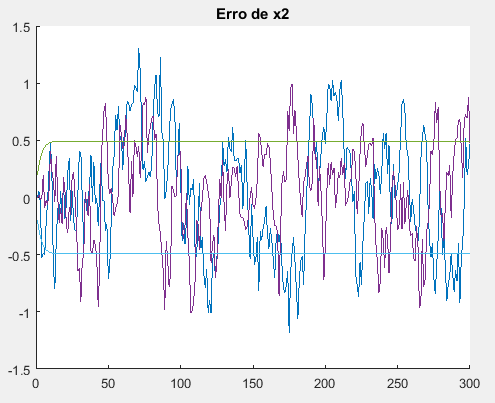
****

**Figura 3**

As figuras 4 e 5 representam o erro de estimação dos parâmetros de X1 e X2 comparados com o desvio padrão das estimações do Filtro de Kalman. Nota-se que a maior parte das estimações dos erros das variáveis esta em conformidade com o seu desvio padrão, assim mostrando a não discrepância dos valores das estimações, logo demonstrando a eficiência do Filtro de Kalman.



**Figura 4**



**Figura 5**