
```

function [X, P, xp, Pp, l] = kalmanFilterextract(Y, x_0, P_0, A, Q, H, R)

%KALMANFILTER Filters measurements sequence Y using a Kalman filter.
%
%Input:
%   Y           [m x N] Measurement sequence
%   x_0         [n x 1] Prior mean
%   P_0         [n x n] Prior covariance
%   A           [n x n] State transition matrix
%   Q           [n x n] Process noise covariance
%   H           [m x n] Measurement model matrix
%   R           [m x m] Measurement noise covariance
%
%Output:
%   x           [n x N] Estimated state vector sequence
%   P           [n x n x N] Filter error covariance
%
```

Parameters

```

N = size(Y,2);

n = length(x_0);
m = size(Y,1);
```

Data allocation

```

X = zeros(n,N);
P = zeros(n,n,N);
n = length(x_0);

xo(:,1)=x_0;
Po(:,:,1)=P_0;
for i=1:N;
%prediction step
xp(:,i)=A*xo(:,i);
Pp(:,:,i)=A*Po(:,:,i)*A'+Q;
%Innovation Covariance
l(i).S=H*Pp(:,:,i)*H'+R;
%innovation
l(i).innov=Y(:,i)-H*xp(:,i);
%Kalman Gain
l(i).Kk=Pp(:,:,i)*H'*inv(l(i).S);
%update step
X(:,i)=xp(:,i)+l(i).Kk*l(i).innov;
P(:, :, i)=Pp(:, :, i)-l(i).Kk*l(i).S*l(i).Kk';
%initialization for next iteration
xo(:,i+1)=X(:,i);
Po(:, :, i+1)=P(:, :, i);
end
end
```

Published with MATLAB® R2020a