Spencer Freeman

AOE 5784, Estimation and Filtering

11/21/2024

Midterm 2

HW4 | Spencer Freeman | AOE 5784 | 11/22/24

4-7) ① Simplify (4.3.3-1) for a time-invariant system, that is, $F(i) = F$, $G(i) = G$.

$$\underline{x}(k) = \left[\prod_{j=0}^{k-\ell-1} F(k-1-j) \right] \underline{x}(\ell) + \sum_{i=\ell}^{k-1} \left(\prod_{j=0}^{k-i-2} F(k-1-j) \right] \cdot \ldots$$

$$[G(i)\underline{u}(i) + \underline{v}(i)]$$

(4.3.3-1)

② Find a close-form solution, similar to 4.3.3-1, for the covariance 4.3.4-7 assuming $F(k) = F$, $\Gamma(k) = \Gamma$, $Q(k) = Q$.

$$P_{xx}(k+1) = F(k) P_{xx}(k) F(k)^T + \Gamma(k) Q(k) \Gamma(k)^T$$

(4.3.4-7)

① $$\prod_{j=0}^{k-\ell-1} F(k-1-j) = F(k-1-(0)) F(k-1-(1)) \cdot \ldots F(k-1-(k-\ell-1))$$

$F = \text{¢} : \hookrightarrow = F^{k-\ell}$

$$\prod_{j=0}^{k-i-2} F(k-1-j) = F(k-1-(0)) \cdot \ldots F(k-1-(k-i-2))$$

$F = \text{¢} : \hookrightarrow = F^{k-i-1}$

$$\boxed{\underline{x}(k) = F^{k-\ell} \underline{x}(\ell) + \sum_{i=\ell}^{k-1} F^{k-i-1} [G\underline{u}(i) + \underline{v}(i)]}$$

②

$$P_{xx}(k+1) = F P_{xx}(k) F^T + \Gamma Q \Gamma^T$$

$$P_{xx}(\overbrace{k+1-1}^{k}) = F P_{xx}(k-1) F^T + \Gamma Q \Gamma^T$$

$$P(k+1) = F\left[F P(k-1) F^T + \Gamma Q \Gamma^T\right] F^T + \Gamma Q \Gamma^T$$

$$P(\overbrace{k+1}^{k}) = F^2 P(\overbrace{k-1}^{l}) F^{T^2} + F^1 \overbrace{\Gamma Q \Gamma^T}^{T} F^{T^1} + \overbrace{\Gamma Q \Gamma^T}^{T}$$

$$\boxed{P(k) = F^{k-l} P(l) F^{T^{k-l}} + \sum_{i=0}^{k-l-1} F^i T F^{T^i}}$$

2) Prove: $\frac{1}{2}\underline{\nu}^T(k+1)S^{-1}(k+1)\underline{\nu}(k+1) = J_b[\hat{\underline{v}}(k), \hat{\underline{x}}(k+1), k]$

$$J_b[\underline{v}(k), \underline{x}(k+1), k] = J_a[F^{-1}(k)(\underline{x}(k+1) - G(k)\underline{u}(k) - \Gamma(k)\underline{v}(k)), \underline{v}(k), \underline{x}(k+1), k]$$

$$J_a[\underline{x}(k), \underline{v}(k), \underline{x}(k+1), k] = \frac{1}{2}[\underline{x}(k) - \hat{\underline{x}}(k)]^T P^{-1}(k)[\underline{x}(k) - \hat{\underline{x}}(k)] + \dots$$

$$\frac{1}{2}\underline{v}^T(k)Q^{-1}(k)\underline{v}(k) + \frac{1}{2}[\underline{z}(k+1) - H(k+1)\underline{x}(k+1)]^T R^{-1}(k+1)[\dots$$

$$\underline{z}(k+1) - H(k+1)\underline{x}(k+1)]$$

$$J_b[\hat{\underline{v}}(k), \hat{\underline{x}}(k+1), k] = \frac{1}{2}\underline{A}^T P^{-1}(k)\underline{A} + \frac{1}{2}\underline{C}^T Q^{-1}(k)\underline{C} + \frac{1}{2}\underline{B}^T R^{-1}(k+1)\underline{B}$$

$$\hat{\underline{x}}(k|k+1) = F^{-1}(k)[\hat{\underline{x}}(k+1) - G(k)\underline{u}(k) - \Gamma(k)\hat{\underline{v}}(k)]$$

$$\underline{A} = [\hat{\underline{x}}(k|k+1) - \hat{\underline{x}}(k)]$$
$$\underline{C} = \hat{\underline{v}}(k)$$
$$\underline{B} = [\underline{z}(k+1) - H(k+1)\hat{\underline{x}}(k+1)]$$

<u>rewrite in terms of innovation, $\underline{\nu}(k+1)$</u>

C $\underline{C} = \hat{\underline{v}}(k) = Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)[\hat{\underline{x}}(k+1) - F(k)\hat{\underline{x}}(k) - G(k)\underline{u}(k)]$

$\bar{\underline{x}}(k+1) = F(k)\hat{\underline{x}}(k) + G(k)\underline{u}(k)$ → state mean propagation

$\hat{\underline{x}}(k+1) = \bar{\underline{x}}(k+1) + \bar{P}(k+1)H^T(k+1)S^{-1}(k+1)\underline{\nu}(k+1)$ → K.F. update

$$\underline{C} = Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)[\bar{\underline{x}}(k+1) + \bar{P}(k+1)H^T(k+1)S^{-1}(k+1)\underline{\nu}(k+1) - \dots$$
$$F(k)F^{-1}(k)(\bar{\underline{x}}(k+1) - G(k)\underline{u}(k)) - G(k)\underline{u}(k)]$$

$$\underline{C} = Q(k)\Gamma^T(k)H^T(k+1)S^{-1}(k+1)\underline{\nu}(k+1)$$

(A) $\underline{A} = F^{-1}(k)[\hat{x}(k+1) - G(k)\underline{u}(k) - \Gamma(k)\hat{\underline{v}}(k)] - \hat{\underline{x}}(k)$

$$\underline{A} = F^{-1}(k)[\bar{\underline{x}}(k+1) + \bar{P}(k+1)H^T(k+1)S^{-1}(k+1)\underline{\nu}(k+1) - \ldots$$
$$G(k)\underline{u}(k) - \Gamma(k)\underline{\hat{v}}(k) - (\bar{\underline{x}}(k+1) - G(k)\underline{u}(k))]$$

$$\underline{A} = F^{-1}(k)[\bar{P}(k+1) - \Gamma(k)Q(k)\Gamma^T(k)]H^T(k+1)S^{-1}(k+1)\underline{\nu}(k+1)$$

$$\bar{P}(k+1) = F(k)P(k)F^T(k) + \Gamma(k)Q(k)\Gamma^T(k) \rightarrow \text{covariance propagation}$$

$$\underline{A} = P(k)F^T(k)H^T(k+1)S^{-1}(k+1)\underline{\nu}(k+1)$$

(B) $\underline{B} = \underline{z}(k) - H(k+1)\hat{\underline{x}}(k+1)$

$$\underline{\nu}(k+1) = \underline{z}(k+1) - H(k+1)\bar{\underline{x}}(k+1)$$

$$\underline{B} = \underline{\nu}(k+1) + H(k+1)\bar{\underline{x}}(k+1) - H(k+1)[\bar{\underline{x}}(k+1) + \ldots$$
$$\bar{P}(k+1)H^T(k+1)S^{-1}(k+1)\underline{\nu}(k+1)]$$

$$\underline{B} = [\underbrace{S(k+1) - H(k+1)\bar{P}(k+1)H^T(k+1)}_{R(k+1)}]S^{-1}(k+1)\underline{\nu}(k+1)$$

Factor Quadratic Terms

$$\underline{A} = \Theta_A S^{-1}(k+1)\underline{\nu}(k+1)$$
$$\underline{B} = \Theta_B S^{-1}(k+1)\underline{\nu}(k+1)$$
$$\underline{C} = \Theta_C S^{-1}(k+1)\underline{\nu}(k+1)$$

$$J_b[\,] = \frac{1}{2}\underline{\nu}(k+1)^T S^{-1}(k+1)^T[\overbrace{\Theta_A^T P^{-1}(k)\Theta_A}^{\Phi_A} + \overbrace{\Theta_C^T Q^{-1}(k)\Theta_C}^{\Phi_C} + \overbrace{\Theta_B^T R^{-1}(k+1)\Theta_B}^{\Phi_B}]\ldots$$
$$S^{-1}(k+1)\underline{\nu}(k+1)$$

$$\phi_A = (P(k)F^T(k)H^T(k+1))^T P^{-1}(k) P(k) F^T(k) H^T(k+1)$$

$$\phi_A = H(k+1)F(k)P^T(k) F^T(k) H^T(k+1)$$

$$\phi_c = H(k+1)\Gamma(k)Q^T(k) Q^{-1}(k) Q(k)\Gamma^T(k)H^T(k+1)$$

$$\phi_c = H(k+1)[\bar{P}(k+1) - F(k)P(k)F^T(k)]^T H^T(k+1)$$

$$\phi_c = H(k+1)\bar{P}^T(k+1)H^T(k+1) - H(k+1)F(k)P^T(k)F^T(k)H^T(k+1)$$

$$\phi_B = R^T(k+1) R^{-1}(k+1) R(k+1)$$

$$\phi_B = R^T(k+1) = S^T(k+1) - H(k+1)\bar{P}^T(k+1)H^T(k+1)$$

Putting it all together:

$$J_b[\,] = \frac{1}{2}\underline{\nu}^T(k+1)S^{-1}(k+1)^T[H(k+1)F(k)P^T(k)F^T(k)H^T(k+1) + \ldots$$
$$H(k+1)\bar{P}^T(k+1)H^T(k+1) + \ldots$$
$$-H(k+1)F(k)P^T(k)F^T(k)H^T(k+1) + \ldots$$
$$S^T(k+1) - H(k+1)\bar{P}^T(k+1)H^T(k+1) \ldots$$
$$]S^{-1}(k+1)\underline{\nu}(k+1)$$

$$J_b[\hat{\underline{v}}(k), \hat{\underline{x}}(k+1), k] = \frac{1}{2}\underline{\nu}^T(k+1)S^{-1}(k+1)^T[S^T(k+1)]S^{-1}(k+1)\underline{\nu}(k+1)$$

$$\boxed{J_b[\hat{\underline{v}}(k), \hat{\underline{x}}(k+1), k] = \frac{1}{2}\underline{\nu}^T(k+1)S^{-1}(k+1)\underline{\nu}(k+1)} \quad \checkmark$$

$$(S(k+1) = S^T(k+1))$$

HW5 Problem 6

Spencer Freeman

AOE 5784

11/21/2024

Results:

HW5-P6

50 Monte Carlo's:

P_obs_10 =

```
  0.001685037889159  -0.001960890248705
 -0.001960890248705   0.132750322569187
```

xtil_mu_10 =

```
  0.009836097018543   0.002413490133935
```

P_est_10 =

```
  0.001684356805503   0.000851637905493
  0.000851637905493   0.089420255095101
```

P_obs_35 =

```
  0.001604954483244   0.000929190452452
  0.000929190452452   0.061380608944991
```

xtil_mu_35 =

```
 -0.001730036769037  -0.082745748260370
```

P_est_35 =

```
  0.001622216524173   0.000662325081077
  0.000662325081077   0.084545653745850
```

1000 Monte Carlo's:

P_obs_10 =

```
  0.001780496800946   0.000613055000826
  0.000613055000826   0.092962887993726
```

xtil_mu_10 =

```
 -0.000920179678361  -0.015401560401631
```

P_est_10 =

```
  0.001684356805503   0.000851637905493
  0.000851637905493   0.089420255095101


P_obs_35 =

  0.001595398953031   0.000386304340128
  0.000386304340128   0.085142434109844


xtil_mu_35 =

  0.000752078684707   0.016454329553106


P_est_35 =

  0.001622216524173   0.000662325081077
  0.000662325081077   0.084545653745850
```

Code:

```
%% Implement a Kalman filter for the example problem that was presented in class
% Spencer Freeman, 11/20/2024
% AOE 5784, Estimation and Filtering
%
% This script solves number 6 of problem set 5
% ------------------------------------------------------------------------
clear;clc;close all

disp('HW5-P6')

format long

%% P3
kf_example02a % bring in data

% Q(k) = 6 and the new measurement noise covariance R(k) = 0.05.
Qk = 6;
Rk = 0.05;

nx = length(xhat0);
n = length(thist) + 1;

nmc = 50; % monte carlo's

disp(" ")
disp("50 Monte Carlo's:")
[P_obs_10, xtil_mu_10, P_est_10, P_obs_35, xtil_mu_35 P_est_35] = run_mc( ...
   thist, Fk,Gammak,Hk,Qk,Rk,xhat0,P0,n, nx, nmc)

nmc = 1000; % monte carlo's

disp(" ")
disp("1000 Monte Carlo's:")
[P_obs_10, xtil_mu_10, P_est_10, P_obs_35, xtil_mu_35 P_est_35] = run_mc( ...
   thist, Fk,Gammak,Hk,Qk,Rk,xhat0,P0,n, nx, nmc)
```

```
function [P_obs_10, xtil_mu_10, P_est_10, P_obs_35, xtil_mu_35 P_est_35] = run_mc( ...
    thist, Fk,Gammak,Hk,Qk,Rk,xhat0,P0,n, nx, nmc)

xtil = nan(nx, n, nmc); % error for MC's

for j = 1:nmc

[xtruehist,zhist] = kf_truthmodel_midterm(Fk,Gammak,Hk,Qk,Rk,xhat0,P0,n - 1);


t = 0; % s
xhat = xhat0; % initial state estimate
phat = P0; % initial state covariance

ts = nan(1, n);
xhats = nan(nx, n);
phats = nan(nx * nx, n);
for i = 1:(n - 1)

    ts(i) = t;
    xhats(:, i) = xhat;
    phats(:, i) = phat(:); % unwrap to column vector

    t = thist(i); % s
    xbar = Fk * xhat; % propagate state estimate
    pbar = Fk * phat * Fk' + Gammak * Qk * Gammak'; % propagate state covariance

    zbar = Hk * xbar; % expected measurement
    z = zhist(i); % actual measurement
    v = z - zbar; % filter innovation

    S = Hk * pbar * Hk' + Rk; % expected measurement covariance
    W = pbar * Hk' * inv(S); % filter gain

    xhat = xbar + W * v; % updated state estimate
    phat = pbar - W * S * W'; % updates state covariance

end % for

% record the final filter outputs
ts(n) = t;
xhats(:, n) = xhat;
phats(:, n) = phat(:); % unwrap to column vector

xtil(:, :, j) = xtruehist' - xhats;

end % for


xtil_10 = squeeze(xtil(:, 10, :))';
xtil_mu_10 = mean(xtil_10, 1);
% P_obs_10 = cov(xtil_10);

temp = nan(2, 2, nmc);
for i = 1:nmc
    temp(:, :, i) = xtil(:, 10, i) * xtil(:, 10, i)';
end
P_obs_10 = mean(temp, 3);

P_est_10 = reshape(phats(:, 10), size(P0));
```

```
xtil_35 = squeeze(xtil(:, 35, :))';
xtil_mu_35 = mean(xtil_35, 1);
% P_obs_35 = cov(xtil_35);

temp = nan(2, 2, nmc);
for i = 1:nmc
   temp(:, :, i) = xtil(:, 35, i) * xtil(:, 35, i)';
end
P_obs_35 = mean(temp, 3);

P_est_35 = reshape(phats(:, 35), size(P0));

end % function
```

Section 3.5

$$\underline{x} = \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix}$$

$$\underline{x}(k+1) = \underbrace{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}_{F} \underline{x}(k) + \underbrace{\begin{bmatrix} T^2/2 \\ T \end{bmatrix}}_{\Gamma} v(k), \qquad k = 0, 1, \dots 99$$

$$\underline{x}(0) = \begin{bmatrix} 0 \\ 10 \end{bmatrix}$$

$$E[v(k)^2] = q$$

$$z(k) = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{H} \underline{x}(k) + w(k), \qquad k = 1, 0, \dots 100$$

$$E[w(k)^2] = r = 1$$

$$w(k) = \phi(k) + c, \qquad \bar{\phi}(k) = 0, \qquad E[\phi(k)^2] = r = 1$$

$$\underline{x} = \begin{bmatrix} \xi \\ \dot{\xi} \\ c \end{bmatrix}$$

1. $$\underline{x}(k+1) = \underbrace{\begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{F} \underline{x}(k) + \underbrace{\begin{bmatrix} T^2/2 \\ T \\ 0 \end{bmatrix}}_{\Gamma} v(k)$$

$$z(k) = \underbrace{\begin{bmatrix} 1 & 0 & 1 \end{bmatrix}}_{H} \underline{x}(k) + \phi(k)$$

2. observability condition: $\begin{bmatrix} H \\ HF \\ \vdots \\ HF^{n_x - 1} \end{bmatrix} \rightarrow$ full rank

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & T & 1 \\ 1 & 2T & 1 \end{bmatrix} \rightarrow \text{rank } 2 < \overset{(3)}{n_x}$$

NOT observable

linearly dependent

HW6 Problem 2

Spencer Freeman

AOE 5784

11/21/2024

Results:

HW6-P2

Filter Problem a

```
xhat(50):
  1.0e+05 *

  -2.1063
  -0.0529
  -0.0007


phat(50):
   8.6366  -0.7391  -6.4165
  -0.7391   1.2292  -0.5956
  -6.4165  -0.5956   8.6089


xhat_SRIF(50):
  1.0e+05 *

  -2.1063
  -0.0529
  -0.0007


phat_SRIF(50):
   8.6366  -0.7391  -6.4165
  -0.7391   1.2292  -0.5956
  -6.4165  -0.5956   8.6089


covariance error metric (50):
  1.0e-09 *

   0.0280   0.2491   0.0196
   0.2716   0.0757   0.0251
   0.0209   0.0249   0.0120
```

Filter Problem b

```
xhat(50):
  1.0e+04 *

  -4.0118
```

```
   0.0254
   0.0022


phat(50):
   6.5794   -0.6665   -5.2465
  -0.6665    0.7258   -0.7852
  -5.2465   -0.7852    6.8169


xhat_SRIF(50):
  1.0e+04 *

  -4.0118
   0.0254
   0.0022


phat_SRIF(50):
   6.5794   -0.6665   -5.2465
  -0.6665    0.7258   -0.7852
  -5.2465   -0.7852    6.8169


covariance error metric (50):
  1.0e-06 *

   0.1397    0.1292    0.1423
   0.1292    0.1412    0.1515
   0.1423    0.1515    0.1444
```

Code:

```
%% Implement a Kalman filter for the example problem that was presented in class
% Spencer Freeman, 11/21/2024
% AOE 5784, Estimation and Filtering
%
% This script solves number 2 of problem set 6
% ------------------------------------------------------------------------
clear;clc;close all

disp('HW6-P2')


%%

kf_example03a % bring in data

[ts, xhats_a, phats_a, ~] = ...
   filter(Fk, Gammak, Hk, Qk, Rk, xhat0, P0, thist, zhist);

[~, xhats_s_a, phats_s_a, ~] = ...
   SRIF(Fk, Gammak, Hk, Qk, Rk, xhat0, P0, thist, zhist);

cov_error_metric_a = abs(phats_a - phats_s_a) ./ (abs(phats_s_a) + eps^6);

kf_example03b % bring in data

[~, xhats_b, phats_b, ~] = ...
   filter(Fk, Gammak, Hk, Qk, Rk, xhat0, P0, thist, zhist);

[~, xhats_s_b, phats_s_b, ~] = ...
   SRIF(Fk, Gammak, Hk, Qk, Rk, xhat0, P0, thist, zhist);

cov_error_metric_b = abs(phats_b - phats_s_b) ./ (abs(phats_s_b) + eps^6);


%% plotting
close all

plot_filter(ts, xhats_a, phats_a, 'KF-a')
plot_filter(ts, xhats_s_a, phats_s_a, 'SRIF-a')
plot_filter(ts, xhats_b, phats_b, 'KF-b')
plot_filter(ts, xhats_s_b, phats_s_b, 'SRIF-b')


%%
ind = 50;

disp(" ")
disp('Filter Problem a')
disp(" ")
disp("xhat(50): ")
disp((xhats_a(:, ind)))
disp(" ")
disp("phat(50): ")
disp((reshape(phats_a(:, ind), size(P0))))
disp(" ")
disp("xhat_SRIF(50): ")
disp((xhats_s_a(:, ind)))
```

```
disp(" ")
disp("phat_SRIF(50): ")
disp((reshape(phats_s_a(:, ind), size(P0))))
disp(" ")
disp("covariance error metric (50): ")
disp((reshape(cov_error_metric_a(:, ind), size(P0))))
disp(" ")


disp("Filter Problem b")
disp(" ")
disp("xhat(50): ")
disp((xhats_b(:, ind)))
disp(" ")
disp("phat(50): ")
disp(reshape(phats_b(:, ind), size(P0)))
disp(" ")
disp("xhat_SRIF(50): ")
disp((xhats_s_b(:, ind)))
disp(" ")
disp("phat_SRIF(50): ")
disp((reshape(phats_s_b(:, ind), size(P0))))
disp(" ")
disp("covariance error metric (50): ")
disp((reshape(cov_error_metric_b(:, ind), size(P0))))
disp(" ")

function plot_filter(ts, xhats, phats, name)

h = figure;
h.WindowStyle = 'Docked';

subplot(3, 1, 1)
plot(ts, xhats(1, :), 'r*'); hold on
plot(ts, xhats(1, :) + sqrt(phats(1, :)) .* [1; -1], 'bo')
grid on
legend('Estimate', '+-1\sigma')
title(name)
ylabel('xhat_1')

subplot(3, 1, 2)
plot(ts, xhats(2, :), 'r*'); hold on
plot(ts, xhats(2, :) + sqrt(phats(5, :)) .* [1; -1], 'bo')
grid on
ylabel('xhat_2')

subplot(3, 1, 3)
plot(ts, xhats(3, :), 'r*'); hold on
plot(ts, xhats(3, :) + sqrt(phats(9, :)) .* [1; -1], 'bo')
grid on
ylabel('xhat_2')

xlabel('Time (s)')

end


function [ts, xhats, phats, evs] = ...
  SRIF(Fk, Gammak, Hk, Qk, Rk, xhat0, P0, thist, zhist)

n = length(thist) + 1;
```

```
nx = length(xhat0);
nv = size(Qk, 1);

t = 0; % s
ev = 0;

Finv = inv(Fk);
info = inv(P0); % information matrix
Rxx = chol(info);
Rvv = inv(chol(Qk))';
Ra = chol(Rk);
Rainv = inv(Ra);
Ha = Rainv' * Hk;
zahist = Rainv' * zhist;
zx = Rxx * xhat0;

G = 0;
u = 0;

xhat = xhat0; % initial state estimate
phat = P0; % initial state covariance

ts = nan(1, n);
xhats = nan(nx, n);
phats = nan(nx * nx, n);
evs = nan(1, n);
for i = 1:(n - 1)

  ts(i) = t;
  xhats(:, i) = xhat;
  phats(:, i) = phat(:); % unwrap to column vector
  evs(i) = ev;

  t = thist(i); % s

  % propagation
  [q, r] = qr([ ...
    Rvv, zeros(nv, nx); ...
    -Rxx * Finv * Gammak, Rxx * Finv]);

  Ta = q';
  Rvvbar = r(1:nv, 1:nv);
  Rvxbar = r(1:nv, nv + 1:end);
  Rxxbar = r(nv + 1:end, nv + 1:end);

  temp = Ta * [zeros(nv, 1); zx];
  zxbar = temp(nv + 1:end, :);

  % measurement update
  [q, r] = qr([Rxxbar; Ha]);

  Tb = q';
  Rxx = r(1:nx, :);

  temp = Tb * [zxbar; zahist(i)];
  zx = temp(1:nx, :);

  % convert back to true states
  Rxxinv = inv(Rxx);
  xhat = Rxxinv * zx;
  phat = Rxxinv * Rxxinv';
```

```
end

% record the final filter outputs
ts(n) = t;
xhats(:, n) = xhat;
phats(:, n) = phat(:); % unwrap to column vector
evs(n) = ev;

end % function

function [ts, xhats, phats, evs] = ...
  filter(Fk, Gammak, Hk, Qk, Rk, xhat0, P0, thist, zhist)

n = length(thist) + 1;
nx = length(xhat0);

t = 0; % s
xhat = xhat0; % initial state estimate
phat = P0; % initial state covariance
ev = 0;

ts = nan(1, n);
xhats = nan(nx, n);
phats = nan(nx * nx, n);
evs = nan(1, n);
for i = 1:(n - 1)

  ts(i) = t;
  xhats(:, i) = xhat;
  phats(:, i) = phat(:); % unwrap to column vector
  evs(i) = ev;

  t = thist(i); % s
  xbar = Fk * xhat; % propagate state estimate
  pbar = Fk * phat * Fk' + Gammak * Qk * Gammak'; % propagate state covariance

  zbar = Hk * xbar; % expected measurement
  z = zhist(i); % actual measurement
  v = z - zbar; % filter innovation

  S = Hk * pbar * Hk' + Rk; % expected measurement covariance
  Sinv = inv(S);
  W = pbar * Hk' * Sinv; % filter gain

  ev = v' * Sinv * v; % estimation error statistic

  xhat = xbar + W * v; % updated state estimate
  phat = pbar - W * S * W'; % updates state covariance

end

% record the final filter outputs
ts(n) = t;
xhats(:, n) = xhat;
phats(:, n) = phat(:); % unwrap to column vector
evs(n) = ev;

end % function
```

HW6 | Spencer Freeman | AOE 5784 | 11/18/24

3)

$$\underline{x}^*(k) = F^{-1}(k)\left[\underline{x}^*(k+1) - G(k)\underline{u}(k) - \Gamma(k)\underline{v}^*(k)\right]$$

equivalent

$$\underline{x}^*(k) = \hat{\underline{x}}(k) + P(k)F^T(k)\bar{P}^{-1}(k+1)\left[\underline{x}^*(k+1) - \bar{\underline{x}}(k+1)\right]$$

A

$$F^{-1}(k)\left[\underline{x}^*(k+1) - G(k)\underline{u}(k) - \Gamma(k)\underline{v}^*(k)\right] - F^{-1}(k)F(k)\hat{\underline{x}}(k) + \hat{\underline{x}}(k)$$

$$F^{-1}(k)\left[\underline{x}^*(k+1) - G(k)\underline{u}(k) - \Gamma(k)\underline{v}^*(k) - F(k)\hat{\underline{x}}(k)\right] + \hat{\underline{x}}(k)$$

$$\underline{v}^*(k) = Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)\left[\underline{x}^*(k+1) - \bar{\underline{x}}(k+1)\right]$$

$\hookrightarrow$ MAP kalman Filter derivation

$$-F(k)\hat{\underline{x}}(k) - G(k)\underline{u}(k) = -\bar{\underline{x}}(k+1) \rightarrow \text{state mean propagation}$$

$$F^{-1}(k)\left[\underline{x}^*(k+1) - \bar{\underline{x}}(k+1) - \Gamma(k)Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)\left(\underline{x}^*(k+1) - \bar{\underline{x}}(k+1)\right)\right] + \hat{\underline{x}}(k)$$

$$F^{-1}(k)\left[\left(I - \Gamma(k)Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)\right)\left(\underline{x}^*(k+1) - \bar{\underline{x}}(k+1)\right)\right] + \hat{\underline{x}}(k)$$

$$\bar{P}(k+1) = F(k)P(k)F^T(k) + \Gamma(k)Q(k)\Gamma^T(k)$$

$\hookrightarrow$ kalman filter covariance propagation

$$I - \left(\bar{P}(k+1) - F(k)P(k)F^T(k)\right)\bar{P}^{-1}(k+1) = +F(k)P(k)F^T(k)\bar{P}^{-1}(k+1)$$

$$F^{-1}(k)\left[F(k)P(k)F^T(k)\bar{P}^{-1}(k+1)\right]\left(\underline{x}^*(k+1) - \bar{\underline{x}}(k+1)\right) + \hat{\underline{x}}(k)$$

$$\boxed{P(k)F^T(k)\bar{P}^{-1}(k+1)\left[\underline{x}^*(k+1) - \bar{\underline{x}}(k+1)\right] + \hat{\underline{x}}(k) = \underline{x}^*(k) = \text{A}}$$