
```

clear variables;
close all;
clc;

addpath(['..\Functions\']);

rng(43, "twister");

% Load the matrices into the workspace
run('kf_example03a.m');
u = zeros(size(zhist, 1), 1);
G = zeros(length(xhat0), 1);

% Redefine some matrices for the final
Qk = [150, 15; 15, 25];
Rk = 4;

% Kalman Filter
[t_kf, x_kf, P_kf, nis_kf] = kalman_filter(Fk, G, Gammak, Hk, Qk, Rk, u,
zhist, xhat0, P0);

% Square-root Information Filter
[t_srif, x_srif, P_srif, nis_srif] = srif(Fk, Gammak, Hk, Qk, Rk, xhat0, P0,
zhist);

% Square-root Information Smoother
[t_sris, x_sris, P_sris, ~, ~] = sris(Fk, Gammak, Hk, Qk, Rk, xhat0, P0,
zhist);

```

Plotting

```

plot_kf(t_kf, x_kf, P_kf, [], [], 3);
plot_kf(t_srif, x_srif, P_srif, [], [], 3);
plot_kf(t_sris, x_sris, P_sris, [], [], 3);

% Print requested values
format long;
xhat10_kf = x_kf(10, :);
P10_kf = P_kf(:, :, 10);
xhat10_srif = x_srif(10, :);
P10_srif = P_srif(:, :, 10);
xstar10 = x_sris(10, :);
Pstar10 = P_sris(:, :, 10);

better_element_wise = Pstar10 < P10_kf
better_RMS = trace(Pstar10) < trace(P10_kf)
better_norm = norm(Pstar10) < norm(P10_kf)

% The scaling on states 1 and 2 makes it harder to see, but state 3 is much
% smoother after the smoothing.

```

```
better_element_wise =
```

```
3×3 logical array
```

```
1 1 1  
1 1 1  
1 1 1
```

```
better_RMS =
```

```
logical
```

```
1
```

```
better_norm =
```

```
logical
```

```
1
```



