
Final Exam Problem 2

Table of Contents

Initialize	1
Batch	2

Initialize

```
clearvars -except function_list pub_opt
global function_list;
function_list = {};
close all

w1 = 2*pi/709;
w2 = 2*pi/383;
w3 = 2*pi/107;
w4 = 2*pi/13;

H = @(t) [1 t t*t t*t*t t*t*t*t cos(w1*t) cos(w2*t) cos(w3*t) cos(w4*t) ...
          sin(w1*t) sin(w2*t) sin(w3*t) sin(w4*t)];

num_state = 13;
obs_data = load('P2_obs.txt');
y = obs_data(:,2);
t = obs_data(:,1);
num_obs = length(y);
H_mat = zeros(num_obs,num_state);
for ii = 1:num_obs
    % t = 0:99
    H_mat(ii,:) = H(t(ii));
end

x_ap = zeros(num_state,1);
P_ap = eye(num_state)*100;
L = chol(P_ap,'lower');
R_ap = eye(num_state)/(L);
b_ap = R_ap*x_ap;

W = 1; % 1/(measurements w/ unit variance)
W_sqrt = sqrt(W);

big_mat = [R_ap; H_mat];
[rows, cols] = size(big_mat);
Q = eye(rows);
R = big_mat;
I = eye(rows);
for ii = 1:cols
```

```

        ii
        for jj = ii+1:rows
            G = I;
            %         theta = atan2(R(jj,ii),R(jj-1,ii));
            %         S = sin(theta);
            temp = sqrt(R(ii,ii)*R(ii,ii)+R(jj,ii)*R(jj,ii));
            S = R(jj,ii)/temp;
            %         C = cos(theta);
            C = R(ii,ii)/temp;
            G(ii,ii) = C;
            G(jj,ii) = -S;
            G(ii,jj) = S;
            G(jj,jj) = C;

            R(ii,:) = [C S]*[R(ii,:);R(jj,:)];
            R(jj,:) = [-S C]*[R(ii,:);R(jj,:)];
            Q(ii,:) = [C S]*[Q(ii,:);Q(jj,:)];
            Q(jj,:) = [-S C]*[Q(ii,:);Q(jj,:)];

            %         R = G*R;
            %         Q = G*Q;
        end
    end
    R = R(1:num_state,1:num_state);

    b_e = Q*W_sqrt*[b_ap;y];
    b = b_e(1:num_state);

    x_est_Givens = R\b;
    P_Givens = eye(num_state)/R/R';
    P_G_diag = diag(P_Givens);
    % A1 is index 6, B1 is 10
    A1 = 6; B1 = 10;
    P_A1_B1 = [P_G_diag(A1) P_Givens(A1,B1); P_Givens(A1,B1) P_G_diag(B1)];
    [evec,ev]=eig(P_A1_B1);
    ell_a=3*sqrt(ev(2,2)); % larger eigenvalue
    ell_b=3*sqrt(ev(1,1));
    angle=atan2(evec(2,2),evec(1,2)); % Using 2nd eigenvector
    figure
    plot_ellipse(ell_a,ell_b,angle,x_est_Givens(A1),x_est_Givens(B1));
    title('A1-B1 3-sigma Probability Ellipse')
    xlabel('A1'),ylabel('B1')

```

Batch

Begin batch

```

chol_P0 = chol(P_ap,'lower');
P0_inv = eye(13)/(chol_P0')/(chol_P0);
info_mat = P0_inv;
norm_mat = P0_inv*x_ap;
for ii = 1:num_obs
    % No integration step with constants. STM = I

```

```

%H
H_ = H(t(ii));

% Accumulate information matrix
info_mat = info_mat + H_'*W*H_;

% Accumulate normal matrix
norm_mat = norm_mat + H_'*W*y(ii);

end
x_est_batch = cholesky_linear_solver(info_mat,norm_mat);
chol_info = chol(info_mat,'lower');
P_batch = eye(13)/(chol_info')/(chol_info);
% P_batch = eye(num_state)/info_mat;
P_b_diag = diag(P_batch);

% Ellipsoid
P_A1_B1 = [P_b_diag(A1) P_batch(A1,B1); P_batch(A1,B1) P_b_diag(B1)];
[evect,ev]=eig(P_A1_B1);
ell_a=3*sqrt(ev(2,2)); % larger eigenvalue
ell_b=3*sqrt(ev(1,1));
angle=atan2(evect(2,2),evect(1,2)); % Using 2nd eigenvector
hold on
plot_ellipse(ell_a,ell_b,angle,x_est_batch(A1),x_est_batch(B1),'r');
legend('Givens','Batch')

```

Published with MATLAB® R2013b