
```

clear variables;
close all;
clc;

addpath('..\Functions\');

% Case i
rbari = 76; thetabari = -3*pi/180;
sr2_i = 1^2; st2_i = (pi/180)^2;
xbari = [rbari; thetabari];
P_pi = diag([sr2_i, st2_i]);

% Case ii
rbarii = 76; thetabarii = -3*pi/180;
sr2_ii = 1^2; st2_ii = (15*pi/180)^2;
xbarii = [rbarii; thetabarii];
P_pii = diag([sr2_ii, st2_ii]);

% Linear transformations
cbar_li = h_nl(rbari, thetabari) % Linearly transformed mean
Pcc_li = H(rbari, thetabari)*P_pi*H(rbari, thetabari).' % Linearly
transformed cov

cbar_lij = h_nl(rbarii, thetabarii) % Linearly transformed mean
Pcc_lij = H(rbarii, thetabarii)*P_pii*H(rbarii, thetabarii).' % Linearly
transformed cov

% UT transformations
nl_handle = @(x) h_nl(x(1), x(2));

[cbar_uti, Pcc_uti] = ut(xbari, P_pi, nl_handle, 1e-3, 2, 0) % Unscented
mean and cov
[cbar_utii, Pcc_utii] = ut(xbarii, P_pii, nl_handle, 1e-3, 2, 0) % Unscented
mean and cov

% Validation with large numbers
% Case i
polar_points_i = draw_gaussian(xbari, P_pi, 1e6);
cartesian_points_i = zeros(2, 1e6);
for i=1:1e6
    cartesian_points_i(:, i) = h_nl(polar_points_i(1, i), polar_points_i(2,
i));
end
cartesian_mean_i = mean(cartesian_points_i, 2) % True mean
cartesian_cov_i = cov(cartesian_points_i.') % True cov

% Case 2
polar_points_ii = draw_gaussian(xbarii, P_pii, 1e6);
cartesian_points_ii = zeros(2, 1e6);
for i=1:1e6
    cartesian_points_ii(:, i) = h_nl(polar_points_ii(1, i),
polar_points_ii(2, i));

```

```

end
cartesian_mean_ii = mean(cartesian_points_ii, 2) % True mean
cartesian_cov_ii = cov(cartesian_points_ii.') % True cov

% Print results
format long;
disp('Case i')
disp('Linearized mean and covariance')
cbar_li = h_nl(rbari, thetabari) % Linearly transformed mean
Pcc_li = H(rbari, thetabari)*P_pi*H(rbari, thetabari).' % Linearly
transformed cov
disp('Unscented mean and covariance')
[cbar_uti, Pcc_uti] = ut(xbari, P_pi, nl_handle, 1e-3, 2, 0) % Unscented
mean and cov
disp('True mean and covariance')
cartesian_mean_i = mean(cartesian_points_i, 2) % True mean
cartesian_cov_i = cov(cartesian_points_i.') % True cov

disp('Case ii')
disp('Linearized mean and covariance')
cbar_lii = h_nl(rbarii, thetabarii) % Linearly transformed mean
Pcc_lii = H(rbarii, thetabarii)*P_pii*H(rbarii, thetabarii).' % Linearly
transformed cov
disp('Unscented mean and covariance')
[cbar_utii, Pcc_utii] = ut(xbarii, P_pii, nl_handle, 1e-3, 2, 0) % Unscented
mean and cov
disp('True mean and covariance')
cartesian_mean_ii = mean(cartesian_points_ii, 2) % True mean
cartesian_cov_ii = cov(cartesian_points_ii.') % True cov

plot_cov_comparison(cartesian_mean_i, cartesian_cov_i, cbar_li, Pcc_li,
cbar_uti, Pcc_uti);
plot_cov_comparison(cartesian_mean_ii, cartesian_cov_ii, cbar_lii, Pcc_lii,
cbar_utii, Pcc_utii)

```

Functions

Transformations

```

function [cartesian] = h_nl(r, theta)
    x = r*cos(theta);
    y = r*sin(theta);
    cartesian = [x; y];
end

function [H] = H(rbar, thetabar)
    H = [cos(thetabar), -rbar*sin(thetabar); sin(thetabar),
rbar*cos(thetabar)];
end

function [cartesian] = H_lin(rbar, thetabar, r, theta)
    H(rbar, thetabar);
    cartesian = h_nl(rbar, thetabar) + H*[r-rbar; theta-thetabar];

```

```

end

% Unscented Transform
function [xbar_t, Pxx_t] = ut(xbar, Pxx, nl_fun, alpha, beta, kappa,
varargin)

    % varargin to take more points

    nx = length(xbar);
    npts = nx; % Doesn't include the mean point and includes symmetry
    Sx = chol(Pxx).';
    lambda = (alpha^2)*(nx+kappa) - nx;

    % Build points
    chi = zeros(nx, 2*npts+1);
    chi0 = xbar;
    chi(:, 1) = chi0;
    for i=1:npts
        chi(:, 2*i) = chi0 + sqrt(nx + lambda)*Sx(:,i);
        chi(:, 2*i+1) = chi0 - sqrt(nx + lambda)*Sx(:,i);
    end

    % Push points through function
    chi_t = zeros(nx, 2*npts+1);
    for m=1:2*npts+1
        chi_t(:, m) = nl_fun(chi(:, m));
    end

    w0 = lambda/(nx+lambda);
    wi = 1/(2*(nx+lambda));

    xbar_t = w0*chi_t(:, 1) + sum(wi*chi_t(:, 2:end), 2);
    Pxx_t = (lambda/(nx+lambda) + 1 - alpha^2 + beta)*(chi_t(:, 1) -
xbar_t)*(chi_t(:, 1) - xbar_t).';
    for k=2:2*npts+1
        Pxx_t = Pxx_t + wi*(chi_t(:, k) - xbar_t)*(chi_t(:, k) - xbar_t).';
    end
end

function plot_cov_comparison(mu_true, P_true, mu_lin, P_lin, mu_ut, P_ut)
% PLOT_COV_COMPARISON
% Visualizes and compares 4 Gaussian distributions.

    % Create figure
    figure;

    hold on; grid on; axis equal;

    % --- 2. True Posterior (Green) ---
    plot_gaussian(mu_true, P_true, 'g', '--', 'x');

    % --- 3. Linearized/EKF (Red) ---
    plot_gaussian(mu_lin, P_lin, 'r', '-', 'o');

```

```

% --- 4. Unscented/UKF (Blue) ---
plot_gaussian(mu_ut, P_ut, 'b', '-.', '^');

legend('True Posterior', 'Linearized (EKF)', 'Unscented (UKF)');
title('Comparison of Mean and Covariance Estimates');
xlabel('X'); ylabel('Y');
set(gca, 'FontSize', 12);
hold off;
end

function plot_gaussian(mu, P, col, style, marker)

% Settings
n_sig = 3;      % 3-sigma
lw      = 2;      % Line Width
ms      = 10;     % Marker Size

mu = mu(:);      % Ensure column vector

% --- 2. Calculate Ellipse Geometry ---
t = linspace(0, 2*pi, 100);
unit_circle = [cos(t); sin(t)];

[V, D] = eig(P);
D = max(D, 0); % Safety for negative numerical noise
scale_mat = sqrt(D) * n_sig;

ellipse_pts = V * scale_mat * unit_circle;
final_pts = ellipse_pts + mu;

% --- 3. Plotting ---
% Only plot if we are holding the plot (standard usage)
% or if no output is requested (void call)
plot(final_pts(1,:), final_pts(2,:), 'Color', col, 'LineStyle', style,
'LineWidth', lw);
plot(mu(1), mu(2), 'Marker', marker, 'Color', col, 'MarkerFaceColor',
col, 'MarkerSize', ms, 'HandleVisibility','off');
end

cbar_li =

    75.895844641347608
   -3.977532674463731

Pcc_li =

    1.002080228655848    0.039693127323976
    0.039693127323976    1.757389988032180

cbar_lii =

```

75.895844641347608
-3.977532674463731

Pcc_lii =

1.0e+02 *

0.020815991663193 0.206381415338717
0.206381415338717 3.947991995884870

cbar_uti =

75.884285032749176
-3.976926862262189

Pcc_uti =

1.002347478188724 0.039679121390983
0.039679121390982 1.757390721692449

cbar_utii =

73.294935077428818
-3.841224780771881

Pcc_utii =

1.0e+02 *

0.156110670363996 0.199290912250872
0.199290912250144 3.948363412514955

cartesian_mean_i =

75.884753413169051
-3.976692998560023

cartesian_cov_i =

1.000698421540194 0.040630042490086
0.040630042490086 1.758457431770547

cartesian_mean_ii =

73.336393441249015

-3.821738179651297

cartesian_cov_ii =

1.0e+02 *

0.146275041276730	0.186508565500269
0.186508565500269	3.691426431082478

Case i

Linearized mean and covariance

cbar_li =

75.895844641347608
-3.977532674463731

Pcc_li =

1.002080228655848	0.039693127323976
0.039693127323976	1.757389988032180

Unscented mean and covariance

cbar_uti =

75.884285032749176
-3.976926862262189

Pcc_uti =

1.002347478188724	0.039679121390983
0.039679121390982	1.757390721692449

True mean and covariance

cartesian_mean_i =

75.884753413169051
-3.976692998560023

cartesian_cov_i =

1.000698421540194	0.040630042490086
0.040630042490086	1.758457431770547

Case ii

Linearized mean and covariance

cbar_lii =

75.895844641347608
-3.977532674463731

Pcc_lii =

1.0e+02 *

0.020815991663193 0.206381415338717
0.206381415338717 3.947991995884870

Unscented mean and covariance

cbar_utii =

73.294935077428818
-3.841224780771881

Pcc_utii =

1.0e+02 *

0.156110670363996 0.199290912250872
0.199290912250144 3.948363412514955

True mean and covariance

cartesian_mean_ii =

73.336393441249015
-3.821738179651297

cartesian_cov_ii =

1.0e+02 *

0.146275041276730 0.186508565500269
0.186508565500269 3.691426431082478

