```
clear
clc
%Load dataset
load('dataset2.mat')

syms x y th T d_ real
syms u w sigq [2 1] real
syms y_ xyl nl sigr [17 2] real

r_maxs = [5,3,1];

T_val = t(2) - t(1);
K = length(t)-1;

x_hat_0 = [-3;-3;-3];%[x_true(1);y_true(1);th_true(1)];
vars_0 = [1,1,0.1];
P_hat_0 = diag(vars_0);

vs = [v.';om.'];

h = [x + T*cos(th)*(u1+w1); y + T*sin(th)*(u1+w1) ; th + T*(u2+w2)];
h_syms = [x,y,th,u1,u2,w1,w2,T];

x_ = [x,y,th];
F = jacobian(h,x_);
F_syms = [th, u1, w1, T];

w_ = [w1,w2];
w_p = jacobian(h,w_);
w_p_syms = [th,T];

Q = diag([sigq1,sigq2]);

for r_max = r_maxs

    x_hat = zeros(3,K+1);
    vars = zeros(3,K+1);
    x_hat(:,1) = x_hat_0;
    vars(:,1) = vars_0;

    x_hat_k = x_hat_0;
    P_hat_k = P_hat_0;
    for k = 1:K
        disp((k/K)*100);
        r_k = r(k,:);
        b_k = b(k,:);
        cond_1 = r_k ~= 0;
        cond_2 = r_k < r_max;
        r_k_valid = cond_1 & cond_2;
        r_k_new = r_k(r_k_valid);
```

```
        b_k_new = b_k(r_k_valid);
        r_k_valid2 = repmat(r_k_valid.',1,2);

        l_vals = l(r_k_valid2);
        y_vals = [r_k_new.',b_k_new.'];
        y_vals_ordered = reshape(y_vals.',[],1);

        y_syms = y_(r_k_valid2);
        l_syms = xyl(r_k_valid2);
        nl_syms = nl(r_k_valid2);
        sigr_syms = sigr(r_k_valid2);
        nl_vals = zeros(length(l_vals),1);

        y_syms_ordered = reshape(reshape(y_syms,[],2).',[],1);
        nl_syms_ordered = reshape(reshape(nl_syms,[],2).',[],1);
        sigr_syms_ordered = reshape(reshape(sigr_syms,[],2).',[],1);

        L = size(l_vals,1)/2;

        g = sym(zeros(2*L,1));
        g_syms = [x_,d_,l_syms.',nl_syms.'];

        for i = 1:L
            gli = [sqrt((l_syms(i) - x - d_*cos(th))^2 + (l_syms(L+i) - y - d_*sin(th))↙
^2) + nl_syms(i) ; atan2((l_syms(L+i) - y - d_*sin(th)), (l_syms(i) - x - d_*cos(th)))↙
- th + nl_syms(L+i)];
            g((2*i-1):2*i) = gli;
        end

        R = diag(sigr_syms_ordered);

        G_ = jacobian(g,x_);
        G__syms = g_syms;

        nl_p = jacobian(g,nl_syms_ordered);

        h_vals = [x_hat_k.',vs(:,k).',0,0,T_val];
        F_vals = [x_hat_k(3),vs(1,k),0,T_val];

        g_vals = [x_hat_k.', d, l_vals.',nl_vals.'];
        G__vals = g_vals;

        Q_p = w_p*Q*w_p.';
        Q_p_syms = [w_p_syms,sigq.'];
        Q_p_vals = [x_hat_k(3),T_val,r_var,b_var];

        sigr_vals = [v_var*ones(L,1),om_var*ones(L,1)];
        sigr_vals_ordered = reshape(sigr_vals.',[],1);

        R_p = nl_p*R*nl_p.';
```

```
        R_p_syms = sigr_syms_ordered;
        R_p_vals = sigr_vals_ordered;


        F_k_1 = double(subs(F,F_syms,F_vals));
        Qp = double(subs(Q_p,Q_p_syms,Q_p_vals));
        h_k = double(subs(h,h_syms,h_vals));
        G_k = double(subs(G_,G__syms,G__vals));
        Rp = double(subs(R_p,R_p_syms,R_p_vals));
        y_k = double(subs(y_syms_ordered,y_syms_ordered.',y_vals_ordered.'));
        g_k = double(subs(g,g_syms,g_vals));

        P_check_k = F_k_1*P_hat_k*F_k_1.' + Qp;
        x_check_k = h_k;
        x_check_k(3) = wrapToPi(x_check_k(3));

        K_k = (P_check_k*G_k.')/(G_k*P_check_k*G_k.' + Rp);
        P_hat_k = (eye(3)-K_k*G_k)*P_check_k;

        innovation = y_k-g_k;
        innovation = wrapinno(innovation);

        x_hat_k = x_check_k + K_k*innovation;
        x_hat_k(3) = wrapToPi(x_hat_k(3));

        x_hat(:,k+1) = x_hat_k;
        vars(:,k+1) = diag(P_hat_k);
    end
    stds = sqrt(vars);

    % Difference between predicted and true state
    x_diff = x_hat(1,:).' - x_true;
    y_diff = x_hat(2,:).' - y_true;
    th_diff = x_hat(3,:).' - th_true;

    % Upper and lower bounds of standard deviation (+/- 3 std deviations)
    x_upper = 3*stds(1,:).';
    x_lower = -3*stds(1,:).';

    y_upper = 3*stds(2,:).';
    y_lower = -3*stds(2,:).';

    th_upper = 3*stds(3,:).';
    th_lower = -3*stds(3,:).';

    % Plot difference between predicted and true state

    figure
    plot(t,x_upper,':',t,x_lower,':',t,x_diff,'-');
    title(["xhatk - xk as a Function of Time for rmax = ", num2str(r_max)]);
```

```
xlabel("Time t (s)");
ylabel("xhatk - xk (m)");
%ylim([-0.2,0.2]);
legend(["+3 Standard Deviations", "-3 Standard Deviations","xhatk - xk"]);
fname = sprintf('Plotxneg3_uz%d.png', r_max);
saveas(gcf,fname)

figure
plot(t,x_upper,':',t,x_lower,':',t,x_diff,'-');
title(["xhatk - xk as a Function of Time for rmax = ", num2str(r_max)]);
xlabel("Time t (s)");
ylabel("xhatk - xk (m)");
ylim([-0.2,0.2]);
legend(["+3 Standard Deviations", "-3 Standard Deviations","xhatk - xk"]);
fname = sprintf('Plotxneg3_z%d.png', r_max);
saveas(gcf,fname)

figure
plot(t,y_upper,':',t,y_lower,':',t,y_diff,'-');
title(["yhatk - yk as a Function of Time for rmax = ", num2str(r_max)]);
xlabel("Time t (s)");
ylabel("yhatk - yk (m)");
%ylim([-0.2,0.2]);
legend(["+3 Standard Deviations", "-3 Standard Deviations","yhatk - yk"]);
fname = sprintf('Plotyneg3_uz%d.png', r_max);
saveas(gcf,fname)

figure
plot(t,y_upper,':',t,y_lower,':',t,y_diff,'-');
title(["yhatk - yk as a Function of Time for rmax = ", num2str(r_max)]);
xlabel("Time t (s)");
ylabel("yhatk - yk (m)");
ylim([-0.2,0.2]);
legend(["+3 Standard Deviations", "-3 Standard Deviations","yhatk - yk"]);
fname = sprintf('Plotyneg3_z%d.png', r_max);
saveas(gcf,fname)

figure
plot(t,th_upper,':',t,th_lower,':',t,th_diff,'-');
title(["thetahatk - thetak as a Function of Time for rmax = ", num2str(r_max)]);
xlabel("Time t (s)");
ylabel("thetahatk - thetak (rad)");
%ylim([-0.2,0.2]);
legend(["+3 Standard Deviations", "-3 Standard Deviations","thetahatk - thetak"]);
fname = sprintf('Plotthetaneg3_uz%d.png', r_max);
saveas(gcf,fname)

figure
plot(t,th_upper,':',t,th_lower,':',t,th_diff,'-');
title(["thetahatk - thetak as a Function of Time for rmax = ", num2str(r_max)]);
```

```
    xlabel("Time t (s)");
    ylabel("thetahatk - thetak (rad)");
    ylim([-0.2,0.2]);
    legend(["+3 Standard Deviations", "-3 Standard Deviations","thetahatk - thetak"]);
    fname = sprintf('Plotthetaneg3_z%d.png', r_max);
    saveas(gcf,fname)

end


fig = figure;
num_frames = K+1;

vid = VideoWriter('animation.avi');
vid.FrameRate = 1/T_val;
open(vid);

for frame = 1:num_frames

    a=3*stds(1,frame); % horizontal radius
    b=3*stds(2,frame); % vertical radius
    x0=x_hat(1,frame); % x0,y0 ellipse centre coordinates
    y0=x_hat(2,frame);
    x0true=x_true(frame);
    y0true=y_true(frame);

    tparam=-pi:0.01:pi;
    x_ellipse=x0+a*cos(tparam);
    y_ellipse=y0+b*sin(tparam);

    scatter(l(:,1).',l(:,2).',100,'k','filled','o');
    hold on
    plot(x_ellipse, y_ellipse, 'r', 'LineWidth', 2);
    plot(x0,y0,'Color', 'r', 'Marker', 'o', 'MarkerFaceColor', 'r', 'LineWidth',1);
    plot(x0true,y0true,'Color', 'b', 'Marker', 'o', 'MarkerFaceColor', 'b',↙
'LineWidth',1);
    hold off

    frame_data = getframe(fig);
    writeVideo(vid, frame_data);
end

close(vid);
```