clear all

load("src\zadanie\_06.mat")

figure

plot(y)

title('Wyjście układu pobudzone szumem')

legend('Sygnał zaszumiony')

xlabel('N')

ylabel('y(t)')

grid on

Znajdowanie minimum L-funkcji:

handle = @(th) l\_fun(th, y, A, B, G, C)

lb = [0,0];

ub = [3,3];

a = [];

b = [];

Aeq = [];

beq = [];

x0 = [0, 0]

[theta\_min, min\_value, ~, ~, ~, ~, H] = fmincon(handle,x0,a,b,Aeq,beq,lb,ub)

Hesjan oraz oszacowanie błędu:

H

S\_theta\_e = inv(H)

Stworzenie wykresu L-funkcji oraz wykresu poziomicowego:

th\_1 = 0:0.1:3;

th\_2 = 0:0.1:3;

Z = zeros(length(th\_1), length(th\_1));

[X,Y] = meshgrid(th\_1,th\_2);

for i = 1:length(th\_1)

for j = 1:length(th\_2)

Z(i, j) = handle([X(i, j), Y(i, j)]);

end

end

figure

hold on

surf(X,Y,Z)

plot3(theta\_min(1), theta\_min(2), min\_value, '.r', MarkerSize=25)

hold off

xlabel('\sigma\_w')

ylabel('\sigma\_v')

zlabel('L-fun')

grid on

figure

contour(X, Y, Z)

title('Wykres poziomicowy L-funkcji')

xlabel('\sigma\_w')

ylabel('\sigma\_v')

grid on

Tworzenie sygnału sterującego:

t = linspace(0, 10, 1000);

n = length(t);

f = 0.5;

u = 20 \* sawtooth(2\*pi\*f\*t);

figure

plot(u)

title('Sygnał sterujący u')

xlabel('t')

ylabel('u(t)')

grid on

Dodanie szumu procesowego, pomiarowego oraz obliczenie wyjścia ze zmiennymi stanu:

x = zeros(2, n);

y2 = zeros(1, n);

w = zeros(2, n);

v = zeros(1, n);

x(1:2, 1) = [0; 0];

for k = 1:n

w(:, k) = theta\_min(1, 1) \* randn(2, 1);

v(k) = theta\_min(1, 2) \* randn(1, 1);

y2(k) = C \* x(:, k) + v(k);

x(:, k + 1) = A \* x(:, k) + B \* u(k) + G\*w(:, k);

end

x = x';

y2 = y2';

figure

subplot(3, 1, 1)

plot(t, y2)

subtitle('Wyjście układu y')

xlabel('t')

ylabel('y(t)')

grid on

subplot(3, 1, 2)

plot(t, x(2:end, 1))

subtitle('Zmienna stanu x\_1')

xlabel('t')

ylabel('x\_{1}(t)')

grid on

subplot(3, 1, 3)

plot(t, x(2:end, 2))

subtitle('Zmienna stanu x\_2')

xlabel('t')

ylabel('x\_{2}(t)')

grid on

Zastosowanie filtru kalmana:

[x\_e, y\_e] = kalman\_filter(theta\_min, y2, u, A, B, G, C);

Uzyskane wyniki oraz błędy:

error\_y = abs(y2 - y\_e);

figure

subplot(2, 1, 1)

hold on

plot(t, y\_e, 'r', LineWidth=2)

plot(t, y2, 'g')

hold off

subtitle('Porównanie sygnału zaszumionego oraz sygnału po filtracji')

legend('Sygnał z filtru kalmana', 'Sygnał zaszumiony')

xlabel('Czas [s]')

ylabel('y(t)')

grid on

subplot(2, 1, 2)

plot(t, error\_y)

subtitle('Wykres błędu sygnału wyjściowego')

legend('Błąd')

xlabel('Czas [s]')

ylabel('$| \hat{y} - y |$', Interpreter='latex')

grid on

error\_x1 = abs(x\_e(:, 1) - x(2:end, 1));

figure

subplot(2, 1, 1)

hold on

plot(t, x(2:end, 1), 'r')

plot(t, x\_e(:, 1), 'g')

hold off

title('Porównanie sygnału zaszumionego oraz sygnału po filtracji')

legend('Sygnał z filtru kalmana', 'Sygnał zaszumiony')

xlabel('Czas')

ylabel('x\_1(t)')

grid on

subplot(2, 1, 2)

plot(t, error\_x1)

subtitle('Wykres błędu pierwszej zmiennej stanu')

legend('Błąd')

xlabel('Czas [s]')

ylabel('$| \hat{x\_1} - x\_1 |$', Interpreter='latex')

grid on

error\_x2 = abs(x\_e(:, 2) - x(2:end, 2));

figure

subplot(2, 1, 1)

hold on

plot(t, x(2:end, 2), 'r', LineWidth=2)

plot(t, x\_e(:, 2), 'g')

hold off

title('Porównanie sygnału zaszumionego oraz sygnału po filtracji')

legend('Sygnał z filtru kalmana', 'Sygnał zaszumiony')

xlabel('Czas')

ylabel('x\_2(t)')

grid on

subplot(2, 1, 2)

plot(t, error\_x1)

subtitle('Wykres błędu pierwszej zmiennej stanu')

legend('Błąd')

xlabel('Czas [s]')

ylabel('$| \hat{x\_2} - x\_2 |$', Interpreter='latex')

grid on

function q=l\_fun(th, y, A, B, G, C)

[nx, ~] = size(A);

N = length(y);

th = th.^2;

q = 0;

m = zeros(nx, 1);

S = zeros(nx, nx);

u = 0;

for k = 1:N - 1

m = A\*m + B\*u;

S = A\*S\*A' + (G\*G') \* th(1);

W = th(2) + C\*S\*C';

e = y(k + 1) - C\*m;

q = (k - 1) \* q/k + (e\*e/W + log(W))/k;

S = S - (S\*(C' \* C)\*S)/W;

m = m+S\*C'\*e/th(2);

end

q = 0.5 \* q;

end

function [x\_e, y\_e] = kalman\_filter(th, y, u, A, B, G, C)

[nx, ~] = size(A);

N = length(y);

th = th.^2;

m = zeros(nx, 1);

S = zeros(nx, nx);

x\_e = zeros(N, nx);

y\_e = zeros(N, 1);

for k = 1:N - 1

m = A\*m + B\*u(k);

S = A\*S\*A' + (G\*G') \* th(1);

W = th(2) + C\*S\*C';

e = y(k + 1) - C\*m;

S = S - (S\*(C' \* C)\*S)/W;

m = m+S\*C'\*e/th(2);

x\_e(k, :) = m;

y\_e(k) = C\*m;

end

end