

```

function Y = cost_function(x,b1,b2,omega,G_data,Kp,Ki)

a1=x(1);
a2=x(2);
a3=x(3);

G = tf([b1,0,b2,0],[1, a1, (a2+Kp*b1), (a3 + Ki*b1), Kp*b2, Ki*b2]);
G = minreal(G);
[gain_est,phase_est] = bode(G,omega);
gain_est = squeeze(gain_est)';
phase_est = squeeze(phase_est)'/180*pi;% deg を rad に変換
G_est=gain_est.*exp(1i*phase_est);% estimate

pole_max = max(real(pole(G)));% 極の実部の最大値が負でなければならない

%Y1 = norm(log10(1 + 10 * abs(G_data - G_est)),2);% 近さ
%Y1 = norm(G_data - G_est,inf);
Y1 = norm((G_data - G_est)./sqrt(abs(G_data)),2);% データが相当きれいな
らば, 重み付き最小二乗

Y2 = 0;
%Y2 = norm(log10(gain_est) - log10(abs(G_data)),1)/length(gain_est);
% Gainの近さ;
%Y2 = norm(log10(gain_est) - log10(abs(G_data)));% Gainの近さ

%% パラメータの妥当性に関して, 安定性を考慮する

% 実験より, 別のゲインの場合でも安定でなければならない
Kp=4; Ki=2;
G = tf([b1,0,b2,0],[1, a1, (a2+Kp*b1), (a3 + Ki*b1), Kp*b2, Ki*b2]);
G = minreal(G);
pole_max2 = max(real(pole(G)));% 極の実部の最大値が負でなければならない

% 実験より, 別のゲインの場合でも安定でなければならない
Kp=10; Ki=10;
G = tf([b1,0,b2,0],[1, a1, (a2+Kp*b1), (a3 + Ki*b1), Kp*b2, Ki*b2]);
G = minreal(G);
pole_max3 = max(real(pole(G)));% 極の実部の最大値が負でなければならない

%% コストの計算
if pole_max >= 0 || pole_max2 >= 0 || pole_max3 >= 0
    Y = Y1 + Y2/100 + 10^10;
else
    Y = Y1 + Y2/100;
end

```

