Uniwersytet Zielonogórski
Instytut Sterowania i Systemów Informatycznych

Laboratorium Teorii Sterowania

Synteza regulatorów minimalizujących normę H_∞

Wprowadzenie

Środowisko Matlab zawiera wiele funkcji umożliwiających analizę i syntezę systemów gdy kryterium optymalności sterowania jest wyrażone przez normę H_∞. Większość potrzebnych funkcji i poleceń, poza oczywiście Control System Toolbox, jest wbudowanych w μ-Analysis and Synthesis Toolbox (mu-tools), Robust Control Toolbox oraz LMI Control Toolbox. W najnowszych wersjach Matlab'a wszystkie potrzebne funkcje są częścią Robust Control Toolbox.

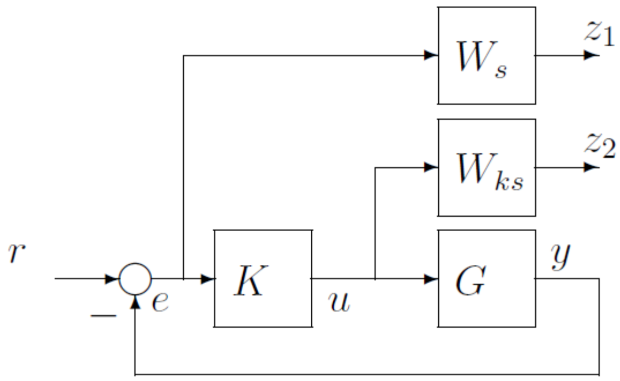
W ogólnym przypadku podstawowy problem syntezy S/KS zostanie zilustrowany poniższym przykładem gdzie zostanie pokazane jak używać podstawowych funkcji w syntezie sterowników H_∞.

Schemat układu regulacji do omawianego problemu jest przedstawiony na Rysunku 1. Transmitancja układu zamkniętego $T = F_l(P, K)$ od w do z może być znaleziona poprzez inspekcję schematu blokowego jako

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} W_s S \\ W_{ks} K S \end{bmatrix} r \tag{1}$$

Uogólniony obiekt regulacji $P(s)$ (zobacz Rysunek 2) może być zapisany jako

$$\left[\begin{array}{c} z_1 \\ z_2 \\ \hline e \end{array}\right] = \left[\begin{array}{c|c} W_s & -W_s G \\ 0 & W_{ks} \\ \hline I & -G \end{array}\right] \begin{bmatrix} r \\ u \end{bmatrix} \tag{2}$$



Rysunek 1: Schemat układu regulacji

Jeśli znamy modele stanowe

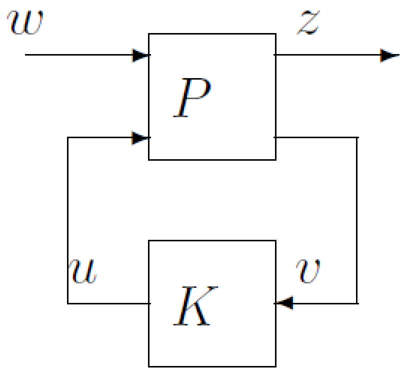
$$G = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \; W_s = \begin{bmatrix} A_s & B_s \\ C_s & D_s \end{bmatrix}, \; W_{ks} = \begin{bmatrix} A_{ks} & B_{ks} \\ C_{ks} & D_{ks} \end{bmatrix}$$

to wtedy może być pokazane, że model stanowy P może być zapisany jako

$$P = \left[\begin{array}{ccc|cc} A_s & 0 & -B_s C & B_s & -B_s D \\ 0 & A_{ks} & 0 & 0 & B_{ks} \\ 0 & 0 & A & 0 & B \\ \hline C_s & 0 & -D_s C & D_s & -D_s D \\ 0 & C_{ks} & 0 & 0 & D_{ks} \\ 0 & 0 & -C & 0 & -D \end{array}\right] \tag{3}$$

Dobór filtrów kształtujących

Filtry kształtujące (tj. wagi) W_s oraz W_{ks} są dobierane przez użytkownika na podstawie wymagań jakościowych regulacji. W typowych sytuacjach potrzebnych jest wiele prób aby dobrać odpowiednie filtry



Rysunek 2: Schemat uogólnionego układu regulacji

kształtujące. W wielu pozycjach literaturowych możemy odnaleźć następującą postać filtrów, które uważana są są za dobry punkt wyjściowy do dalszych prób, tj.

$$W_s = \frac{s/M + \omega_0}{s + \omega_0 A}; \quad W_{ks} = \text{const}$$

gdzie

- ω_0 - pożądana wartość pasma przenoszenia w [rad/sec]. Jest to w przybliżeniu wartość dla której $S(j\omega_0) = -3$[dB].
- A jest minimalną dopuszczalną wartością wzmocnienia S więc odpowiada to dopuszczalnej wartości uchybu regulacji. Typowa wartość A to np. $A = 0.01$ (Pamiętaj, że zawsze $A < 1$)
- M jest maksymalną dopuszczalną wartością $\max(S(j\omega))$. Oznacza to, że dopuszczamy M jako maksymalną wartość funkcji czułości S (transmitancji uchybowej) dla wszystkich częstotliwości. Ta wartość odpowiada z a dobranie odpowiedniego poziomu odporności układu regulacji na niepewności modelowania. Standardowo $M = 2$.

Powyższe wartości pozwalają na odpowiedni dobór kształtu W_s. Tak naprawdę to W_s^{-1} będzie stanowiło górne ograniczenie dla wartości S gdyż zawsze musi być spełniony warunek

$$|S(j\omega)W_s(j\omega)| < 1, \forall\omega \in \mathbb{R} \text{ czyli } |S(j\omega)| < |1/W_s(j\omega)|, \ \forall\omega \in \mathbb{R}$$

Równocześnie W_{ks} jest najczęściej stałą wartością taką, że $u(t) < 1/W_{ks}$. Oznacza to, że $1/W_{ks}$ jest maksymalną dopuszczalną wartością sygnału sterującego ($u(t)$) generowanego przez regulator.

W wielu praktycznych przypadkach, dodatkowo kształtowana jest charakterystyka amplitudowa funkcji $T = GK(I - GK)^{-1}$. Jest to możliwe poprzez dodanie dodatkowego wyjścia $z_3 = W_t y$. Punktem startowym jest wtedy dobór następującego filtra kształtującego

$$W_t = \frac{s + \omega_0/M}{As + \omega_0}$$

który jest symetryczny do W_s względem linii $\omega = \omega_0$

Zapisanie podsystemów układu regulacji

Istnieje wiele możliwych sposobów zapisu dynamicznych układów G, W_s i W_{ks} w środowisku Matlab. Najbardziej popularne to te wykorzystujące polecenia `tf`, `ss` oraz `zpk` (są to polecenia Control System Toolbox). Możliwe jest też użycie bezpośrednio poleceń skrzynki narzędziowej μ-Analysis and Synthesis Toolbox, w szczególności korzystamy z `pck`, `nd2sys` oraz `zp2sys`. Inne funkcje to `mksys` i `tree`. Pamiętaj jednak istnieje zasadnicza różnica pomiędzy zapisami generowanymi przez obie skrzynki narzędziowe.

Wyznaczanie uogólnionego obiektu P

Tworząc uogólniony obiekt regulacji P (czyli transmitancja G wraz z filtrami kształtującymi) mamy wiele możliwości. Oto najpopularniejsze z nich.

1. Zapisz bezpośrednio transmitancję otrzymaną w (2). Przekształcając do modeli stanowych powinno używać się funkcji `minreal` w celu otrzymania minimalnej realizacji modelu (skasowanie tych samych zer i biegunów transmitancji). Użyteczne polecenia to: `sbs`, `abv`, `mmult`, `minv`.

2. Wyznacz bezpośrednio macierze modelu stanowego P tak jak jest to zrobione w (3) i użyj `P=pck(Ap,Bp,Cp,Dp)`.

3. Korzystanie z funkcji `sysic` gdzie specyfikujemy wszystkie podsystem i połączenia pomiędzy nimi.

4. Używanie `sconnect`, gdzie parametrami wejścowymi tej funkcji są podsystemy wejścia i wyjścia, a otrzymujemy jako parametr wyjściowy połączony system.

5. Korzystanie z funkcji `iconnect`, która jest funkcjonalnie podobna do `sysic`

Według różnych opinii, preferowane jest użycie funkcji `sysic` oraz `iconnect` gdyż relatywnie łatwo się ich używa i pozwalają na swobodne tworzenie dowolnych połączeń.

Równocześnie bardzo ważne jest uzyskiwanie zrównoważonej realizacji (ang. *balanced realization*) używanych modeli aby uniknąć ewentualnych problemów numerycznych podczas rozwiązywania problemów optymalizacyjnych sterowania - korzystaj z polecenia `balreal`.

Synteza sterowników

Naszym problemem jest znalezienie takiego sterownika K który stabilizuje nam obiekt regulacji i minimalizuje następującą funkcję celu

$$\|F_l(P,K)\|_\infty = \left\| \begin{array}{c} W_s S \\ W_{ks} K S \end{array} \right\|_\infty$$

Jak można się spodziewać jest wiele metod pozwalających rozwiązać powyższy problem. Najbardziej typowe i popularne metody zostały zaimplementowane w środowisku Matlab jako następujące funkcje

- `hinfsyn` - wyznaczenie sterownika metodą iteracyjną;
- `hinfric` - wyznaczenie sterownika poprzez rozwiązanie równania Riccati'ego;
- `hinflmi` - wyznaczenie sterownika poprzez rozwiązanie liniowego równania macierzowego (LMI).

Wszystkie powyższe funkcje jako jeden z parametrów wymagają modelu uogólnionego obiektu regulacji (obiekt P), który musi zostać utworzony przed wywołaniem tych funkcji (tj. jednej z nich). Dodatkowo w najnowszych wersjach środowiska Matlab mamy możliwość rozwiązania naszego problemu poprzez wywołanie funkcji `mixsyn` gdzie parametrami wejściowymi tej funkcji są G, W_s oraz W_{ks} (i dodatkowo W_t jeśli jest taka potrzeba). Wynika z tego, że nie musimy definiować transmitancji (obiektu) P, gdyż jest on tworzony wewnętrznie przez tą funkcję.

Należy również zaznaczyć, że istniej więcej funkcji pozwalających nasz problem. Przykładem takich funkcji są `ncfsyn` i `loopsyn`, `hinfmix` oraz `msfsyn`. Ich użycie można sprawdzić poprzez wywołanie pomocy.

Analiza wyników

Po uzyskaniu sterownika możemy przejść do ostatniego kroku czyli analizy i walidacji uzyskanych wyników. W tym celu korzystamy ze standardowych funkcji skrzynki narzędziowej Control System Toolbox jak

- `lsim`, `step` - odpowiedzi w dziedzinie czasu;
- `bode`, `freqresp` - odpowiedzi w dziedzinie częstotliwości;
- `sigma` - wykres wartości osobliwych jako funkcji częstotliwości

Podobne funkcje dostępne są w μ-Analysis and Synthesis Toolbox i są to

- `trsp` - odpowiedź w dziedzinie czasu;
- `frsp` - odpowiedź w dziedzinie częstotliwości;
- `vsvd` - wykres wartości osobliwych jako funkcji częstotliwości;
- `vplot` - dowolny wykres w dziedzinie częstotliwości.

Zadania

1. Pokaż analitycznie, że model stanowy zapisany w (3) jest prawidłowy.

2. Rozważmy następujące obiekt regulacji

$$G(s) = \frac{(s-1)}{(s+1)}$$

Znajdź odpowiedni regulator korzystając z funkcji `hinfsyn` lub `mixsyn` gdy przyjmiemy, że

$$W_s = \frac{0.1(s+1)}{100s+1}, \quad , W_{ks} = 0.1 \ \ W_t = []$$

Sprawdź poprawność otrzymanych wyników.

3. Rozważmy następujące obiekt regulacji

$$G(s) = \frac{1}{(s+1)(s+2)}$$

Dla tego obiektu zaprojektuj regulator, tak aby były spełnione następujące wymagania jakościowe regulacji

- pasmo przenoszenia $\omega_B = 3[rad/s]$;
- maksymalny uchyb w stanie ustalonym na poziomie 0.01;
- tolerowanie przez transmitancję układu zamkniętego niepewności rzędu 20 [dB].

Pokaż, że możliwym wyborem filtrów wagowych jest

$$\begin{aligned} W_s &= \frac{s/M + \omega_0}{s + \omega_0 A} = \frac{0.67(s+4.5)}{s+0.003}; \\ W_t &= \frac{s + \omega_0/M}{As + \omega_0} = \frac{100(s+2)}{s+300} \end{aligned}$$

4. Rozważmy następujące obiekt regulacji

$$G(s) = \frac{400}{s^2 + 2s + 400}$$

Dla tego obiektu zaprojektuj regulator, tak aby były spełnione następujące wymagania jakościowe regulacji

- pasmo przenoszenia $\omega_B = 50[rad/s]$;
- $\|KS\|_\infty \leq 15$;
- maksymalny uchyb w stanie ustalonym na poziomie 0.01;
- brak przeregulowania.

5. Rozważmy następujące obiekt regulacji

$$G(s) = \frac{20(s+3)(s-26)}{(s+10)(s-6)(s+20)(s+26)}$$

Dla tego obiektu zaprojektuj regulator, tak aby były spełnione następujące wymagania jakościowe regulacji

- pasmo przenoszenia $\omega_B = 41[rad/s]$;
- $\|KS\|_\infty \leq 70$;
- maksymalny uchyb w stanie ustalonym na poziomie 0.01;
- Przeregulowanie POS=60%.

Skrypt Matlab'a do projektowania regulatorów H_∞

```
% Ten skrypt ilustruje procedure projektowania
% regulatorów minimalizujących normę Hinfty.
% Przykład pokazuje różne sposoby syntezy  regulatora rozwiązującego
% mieszany  problem S/KS z filtrami wagowymi (kształtujacymi).

clear all;
close all;
clc;

%--------------------------------
% Definicje posystemów
%--------------------------------
%Obiekt regulacji  G=200/((10s+1)(0.05s+1)^2)

% Sposob 1, bezpośrednio poprzez mu-tools :
G = nd2sys(1,conv([10,1],conv([0.05 1],[0.05 1])),200);
% Sposob 2, poprzez CST:
% s = tf('s');
% Gcst = 200/((10*s+1)*(0.05*s+1)^2);
% [a,b,c,d] = ssdata(balreal(Gcst));
% G = pck(a,b,c,d);


% Filtry wagowe  Ws = (s/M+w0)/(s+w0*A), Wks=1
M = 1.5;
w0 = 10;
A=1.e-4;
Ws = nd2sys([1/M w0],[1 w0*A]);
Wks = 1;
%--------------------------------
%Budowanie uogólnionego obiektu regulacji  P
%--------------------------------
%Sposób podstawowy:
% /z1\ /Ws -Ws*G\ /r\
% |z2| = |0 Wks | | |
% \ v/ \I -G / \u/
% Reprezentacja w dziedzinie transmitancji
Z1 = sbs(Ws,mmult(-1,Ws,G)); % polecenie sbs słuzy do łączenia podsystemów
Z2 = sbs(0,Wks);             % mmult do mnozenia podstemów
V = sbs(1,mmult(-1,G));
P0 = abv(Z1,Z2,V);           % blokowy zapis transmitancji

%P0 najczesciej nie jest minimalna realizacja wiec zrobmy to
[a,b,c,d] = unpck(P0);
[ab,bb,cb,db] = ssdata( balreal( minreal( ss(a,b,c,d) ) ) );
P0 = pck(ab,bb,cb,db); % pakujemy do odpowiedniego zapisu  jeszcze raz


%--------------------------------
%Budowanie uogólnionego obiektu regulacji  P
%--------------------------------
%Sposób alternatywny:
% /z1\ /W1 -W1*G\ /r\
% |z2| = |0 W2 | | |
% \ v/ \I -G / \u/

%stanowa realizacja podsystemów:
[A,B,C,D] = unpck(G);
[A1,B1,C1,D1] = unpck(Ws);
[A2,B2,C2,D2] = unpck(Wks);
%odczytujemy liczbę wejsc i wyjsc:
n1 = size(A1,1); [q1, p1] = size(D1);
```

```
n2 = size(A2,1); [q2, p2] = size(D2);
n = size(A,1) ; [p , q ] = size(D);
%stanowa reprezentacja P:
Ap = [ A1 , zeros(n1,n2) , -B1*C ;
zeros(n2,n1) , A2 , zeros(n2,n) ;
zeros(n ,n1) , zeros(n ,n2) , A ];
Bp = [ B1 ,-B1*D;
zeros(n2,p) , B2 ;
zeros(n ,p) , B ];
Cp = [ C1 , zeros(q1,n2) , -D1*C ;
zeros(q2,n1), C2 , zeros(q2,n) ;
zeros(q ,n1), zeros(q ,n2) , -C ];
Dp = [ D1 , -D1*D;
zeros(q2,p ), D2 ;
eye(p) , -D ];
%robimy tzw. balanced realization (balreal) aby uniknac pewnych problemow
%numerycznych
[Apb,Bpb,Cpb,Dpb] = ssdata( balreal( ss(Ap,Bp,Cp,Dp) ) );
P1 = pck(Apb,Bpb,Cpb,Dpb);
%---------------------------------
%Budowanie uogólnionego obiektu regulacji  P
%---------------------------------
%Sposób aternatywny 2:

systemnames = 'G Ws Wks';
inputvar = '[r(1); u(1)]';      %wejscia sa skalarami,
outputvar = '[Ws; Wks; r-G]';
input_to_G = '[u]';
input_to_Ws = '[r-G]';
input_to_Wks = '[u]';
sysoutname = 'P2';
cleanupsysic = 'yes';
sysic
%---------------------------------
%Budowanie uogólnionego obiektu regulacji  P
%---------------------------------
%Sposób aternatywny 3:

inputs = 'r(1); u(1)';
outputs = 'Ws; Wks; e=r-G';
K_in = []; %nie mamy jeszcze regulatora (sterownika)
G_in = 'G:u';
Ws_in = 'Ws:e';
Wks_in = 'Wks:u';

[P3,r] = sconnect(inputs,outputs,K_in,G_in,G,Ws_in,Ws,Wks_in,Wks);
%---------------------------------
%Budowanie uogólnionego obiektu regulacji  P
%---------------------------------
%Sposób aternatywny 3: korzystamy z funkcji 'iconnect':
% r = icsignal(1);
% u = icsignal(1);
% ws = icsignal(1);
% wks = icsignal(1);
% e = icsignal(1);
% y = icsignal(1);
% M = iconnect;
% M.Input = [r;u];
% M.Output = [ws;wks;e];
% M.Equation{1} = equate(e,r-y);
% M.Equation{2} = equate(y,ss(A,B,C,D)*u);
% M.Equation{3} = equate(ws,ss(A1,B1,C1,D1)*e);
% M.Equation{4} = equate(wks,ss(A2,B2,C2,D2)*u);
% [ab,bb,cb,db] = ssdata( balreal(M.System) );
```

```
% P4 = pck(ab,bb,cb,db);

%% Synteza regulatora
% Mozemy wybrac sobie dowolna metode tworzenia uogolnionego obiektu
% regulacji. Po prostu P=P0 lub P=P1 lub P=P2 lub P=P3 lub P=P4
P = P1; %(0-4)
% i dobierz parametry optymalizacji
nmeas = 1; nu = 1; gmn=0.5; gmx=20; tol = 0.001;

% mozemy rowniez wybrac sobie metode syntzy regulatora
% czyli wybieramy albo "hinfsyn" lub  "hinflmi" lub "hinfric"

%[K,CL,gopt] = hinfsyn(P,nmeas,nu,gmn,gmx,tol);
[gopt,K] = hinflmi(P,[nmeas, nu],0,tol); CL = starp(P,K,nmeas,nu);
%[gopt,K] = hinfric(P,[nmeas, nu],gmn,gmx); CL = starp(P,K,nmeas,nu);

%Jako alternastywe możemy to wszystko zrobic jak ponizej
% uzyskujac opisy stanowe wszystkich transmitancji

%[a,b,c,d] = unpck(G); Gcst = ss(a,b,c,d);
%[a,b,c,d] = unpck(Ws); Wscst = ss(a,b,c,d);
%[a,b,c,d] = unpck(Wks); Wkscst = ss(a,b,c,d);
%[K,CL,gopt] = mixsyn(Gcst,Wscst,Wkscst,[]);
%[a,b,c,d] = ssdata( balreal(K) ); K = pck(a,b,c,d);
%[a,b,c,d] = ssdata( balreal(CL) ); CL = pck(a,b,c,d);


%% Analiza wyników

w = logspace(-4,6,50);
CLw = vsvd(frsp(CL,w));
figure(1); vplot('liv,m',CLw);
title('singular values of weighted closed loop system');

% Tworzymy podstawowe funkcje transmitacji
[type,out,in,n] = minfo(G);
I = eye(out);
S = minv(madd(I,mmult(G,K))); %f. czulosci (transmitancja uchybowa)
T = msub(I,S); %komplementarna f. czulosci
KS = mmult(K,S); % wyjscie regulatora (wejscie do G)
GK = mmult(G,K); % transmitncja ukladu otwartego

% Wartości osobliwe jako funkcje czestotliwosci
Sw = vsvd(frsp(S,w));
Tw = vsvd(frsp(T,w));
Kw = vsvd(frsp(K,w));
KSw = vsvd(frsp(KS,w));
GKw = vsvd(frsp(GK,w));

figure(2);
vplot('liv,lm',Sw,'-',Tw,'--',GKw,'-.');
title('\sigma(S(jw)) (solid), \sigma(T(jw)) (dashed) and \sigma(GK(jw)) (dashdot)');
xlabel('Frequency [rad/sec]');
ylabel('Amplitude')
figure(3);
vplot('liv,lm',Kw);
title('\sigma(K(jw))');
xlabel('Frequency [rad/sec]');
ylabel('Amplitude')

%% Odpowiedzi czestotliwosciowe
% Sprawdzamy czy w wyniku obiczeń
% dostaliśmy to czego chcieliśmy
```

```
Sd = minv(Ws);
Sdw = vsvd(frsp(Sd,w)); % pożadana funkcja czułości
KSd = minv(Wks);
KSdw = vsvd(frsp(KSd,w)); %pożadane wyjscie regulatora
figure(4);
vplot('liv,lm',Sw,'-',Sdw,'--');
title('\sigma(S(jw)) (solid) and \sigma(Ws^{-1}(jw)) (dashed)');
xlabel('Frequency [rad/sec]');
ylabel('Amplitude')
figure(5);
vplot('liv,lm',KSw,'-',KSdw,'--')
title('\sigma(KS(jw)) (solid) and \sigma(Wks^{-1}(jw)) (dashed)');
xlabel('Frequency [rad/sec]');
ylabel('Amplitude')

%% Odpowiedzi skokowe
reference = 1;
tfinal = 1;
step = 0.01;
y = trsp(T,reference,tfinal,step);
u = trsp(KS,reference,tfinal,step);
figure(6);
subplot(2,1,1);
vplot('iv,d',y);
title('Step response');
ylabel('y');
subplot(2,1,2);
vplot('iv,d',u);
ylabel('u');
xlabel('time');
```