Uniwersytet Zielonogórski
Instytut Sterowania i Systemów Informatycznych

Laboratorium Sterowania Procesami ciągłymi

Sterowanie optymalne przy kwadratowym wskaźniku jakości (LQR)

1. Wprowadzenie

   (a) Problem sterowania optymalnego

   Zakładamy, że system liniowy znajduje się w stanie równowagi. Celem sterowania jest utrzymanie systemu w stanie równowagi - lub w zadanym punkcie pracy - pomimo oddziałujących na niego zakłóceń.
   Zakładamy ponadto, że znany jest opis systemu w przestrzeni stanów oraz jego stan początkowy:

$$\begin{aligned} \dot{x}(t) =& Ax(t) + Bu(t); \quad x(0) = x_0 \\ y(t) =& Cx(t) \end{aligned} \tag{1}$$

   Należy wyznaczyć sterowanie $u = u(t)$, które minimalizuje kwadratowy wskaźnik jakości:

$$J = \frac{1}{2}\int_0^T \left(x^T(t)Qx(t) + u^T(t)Ru(t)\right) dt \tag{2}$$

   gdzie Q jest macierzą symetryczną dodatnio półokreśloną, R jest macierzą symetryczną dodatnio określoną.

   (b) Dobór wartości elementów macierzy Q i R.
   W praktyce dobieramy jedynie wartości elementów na diagonali macierzy Q i R w następujący sposób:
   - $Q_{ii} = 1/$ max. akceptowalna wartość $[x_i^2]$
   - $R_{ii} = 1/$ max. akceptowalna wartość $[u_i^2]$

   (c) Instrukcje Matlab'a.
   Rozwiązanie problemu możemy wyznaczyć przy użyciu poleceń `lqr` oraz `lqry`. Wywołujemy je w następujący sposób:
   $[K, P, ev] = lqr(A, B, Q, R, N)$
   $[K, P, ev] = lqr(A, B, C, D, Q, R)$
   gdzie
   - K macierz wzmocnień sterownika
   - P macierz Lapunowa
   - ev jest wektorem wartości własnych układu zamkniętego;
   - N jest dodatkową macierzą f. kosztu (opcjonalna)

   Polecenie `lqry` rozwiązuje przypadek specjalny gdzie ważone są wyjścia zamiast zmiennych stanu (czyli w funkcji kosztu mamy $y^T(T)Qy(t) + u^T(t)Ru(t)$). Dyskretna wersja polecenia *lqr* to *dlqr*. Algebraiczne Równanie Riccati'ego możemy rozwiązać przy pomocy polecenia `are` (`dare` dla wersji dyskretnej), równanie postaci:

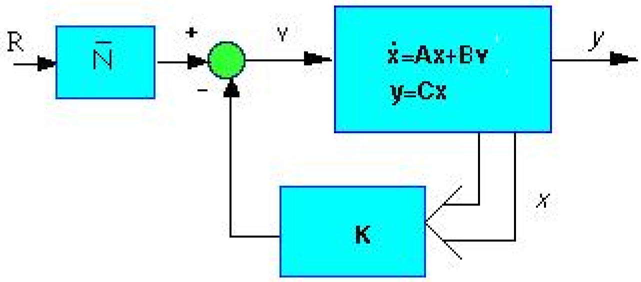
$$A^T X + XA + XBX + C = 0$$

   rozwiązujemy w Matlab'ie tak: `are(A,B,C)`. Należy jednak pamiętać że macierz B musi być symetryczna i nieujemnie określona, a macierz C symetryczna

2. Dane jest model wahadła odwróconego opisanego w przestrzeni stanów następującymi równaniami

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.1818 & 2.6727 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.4545 & 31.1818 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 1.8182 \\ 0 \\ 4.5455 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \end{aligned} \tag{3}$$

   gdzie:



Rysunek 1: Schemat układu regulacji

- x - położenie wózka na którym zamocowane jest wahadło
- θ - kąt odchylenia wahadła od pozycji horyzontalnej

(a) Zakładając dostępność pomiarową stanu, zaprojektuj sterownik minimalizujący kwadratowy wskaźnik jakości, przyjmując $R = 1$ i $Q = C^T C$ czyli

$$Q = \begin{bmatrix} z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

gdzie $z = 1$ i $y = 1$. Nastepnie przeprowadź symulacje układu wahadła (polecenie `lsim`) gdy sygnałem wejściowym jest sygnał stały $= 0.2$ (czyli ustawienie pozycji wahadła na pozycji 0.2m). Sprawdź czy wózek wahadła ustawi się na założonej pozycji. Co więcej, na wykresie charakterystyke kąta odchylenia wahadła (czyli θ, które jest jedna ze zmiennych stanu). Co zaobserwowałeś?

(b) Zbadaj wpływ zmian wartości parametrów z i y i sprawdź czy można polepszyć jakość sterowania (ponownie wykreśl przebiegi symulacji układu)

(c) Wyznacz takie z i y tak aby zostały spełnione następujące wymagania

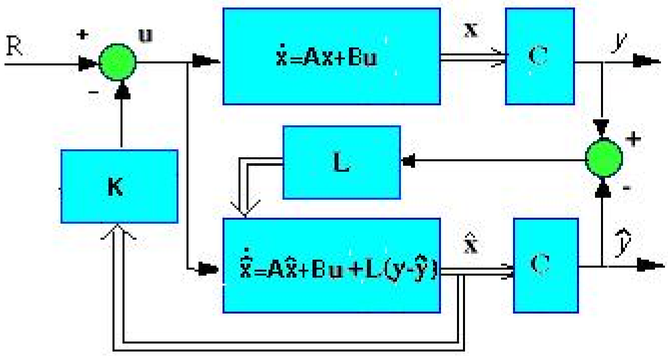
- czas ustalania dla x i θ mniejszy niż 5 sec.
- czas narastania dla x mniejszy niż 1 sec.
- Przeregulowanie dla θ mniejsze niż 20 stopni (0.35 rad).
- Uchyb w stanie ustalonym mniejszy niż 2%.

(d) W celu uzyskania odpowiedniej wartości wyściowej (a dzięki temu uzyskanie pożądanej wartości uchybu regulacji) musimy dokonać rekonfiguracji rozpatrywanego układu regulacji do postaci przedstawionej na rysunku 1. Poszukiwana macierz $\overline{N}$ może być wyznaczona przy użyciu funkcji `rscale` (jej kod znajduje się w ostatniej sekcji tej instrukcji). Można to zrobić wykorzystując poniższy fragment kodu

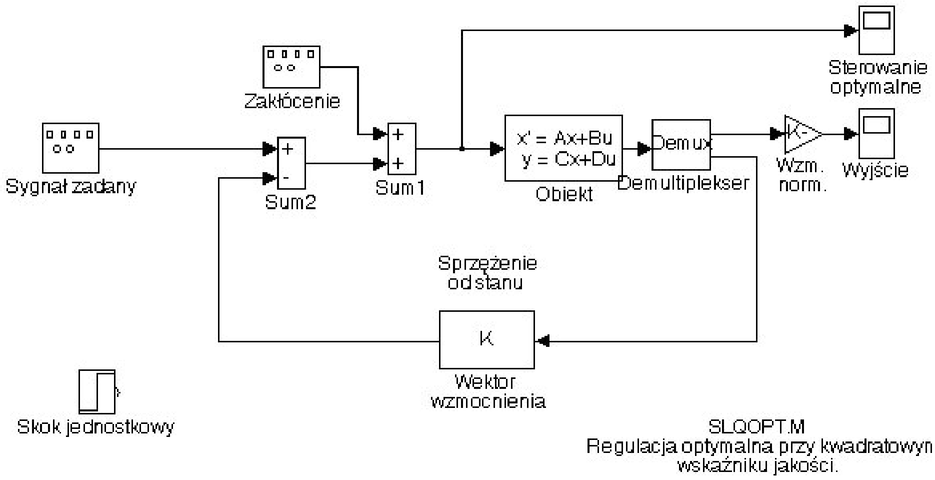
```
Cn=[1 0 0 0];
Nbar=rscale(A,B,Cn,0,K);
Bcn=[Nbar*B];
[Y,X]=lsim(Ac,Bcn,Cc,Dc,U,T);
plot(T,Y)
legend('Wózek','Wahadło')
```

(e) Zakładając, że stan układu nie jest dostępny pomiarowo należy zaprojektować układ regulacji z obserwatorem stanu - patrz rysunek 2. Sprawdź położenie biegunów układu zamkniętego i zaprojektuj obserwator stanu tak jego bieguny były 4 razy szybsze niż bieguny sterownika. Pamiętaj że używamy obu wyjść (kąt odchylenia wahadła i pozycja wózka) przy projektowaniu obserwatora Jednak układ nie jest obserwowalny tylko w przypadku obserwacji kąta odchylenia wahadła (możesz to sprawdzić poleceniem `rank(obsv(A,C(2,:)))`). Ma to oczywiście praktyczną interpretacje: w przypadku pomiaru kąta odchylenia wahadła nie możemy mierzyć położenia wózka i odwrotnie Zbuduj układ regulacji z obserwatorem. narysuj wykresy położenia wózka i kąta odchylenia wahadła. Skorzystaj z następującego fragmentu kodu

```
Ace = [A-B*K              B*K;
        zeros(size(A)) (A-L*C)];
```



Rysunek 2: Schemat układu regulacji z obserwatorem stanu



Rysunek 3: Schemat układu

```
Bce = [       B*Nbar;
      zeros(size(B))];
Cce = [Cc zeros(size(Cc))];
Dce = [0;0];
T = 0:0.01:5;
U = 0.2*ones(size(T));
[Y,X] = lsim(Ace,Bce,Cce,Dce,U,T);
plot(T,Y)
legend('Wózek','Wahadło')
```

3. **Projektowanie układu zamkniętego z regulatorem optymalnym**

   (a) Przeanalizować program `lqopt` projektowania optymalnego układu regulacji. Przed uruchomieniem programu należy wprowadzić do przestrzeni roboczej macierze `A`,`B`,`C`,`D`. Po każdorazowym uruchomieniu programu syntezy układu sterowania optymalnego `lqopt` uruchomić program symulacji układu `slqopt` (schemat blokowy układu przedstawiono na Rys. 3, a poniżej podano parametry symulacji).

   (b) Zbadać wpływ zmian wartości elementów diagonalnych macierzy `R` (parametr `r` programu) na wyniki regulacji.

   (c) Zbadać wpływ zmian wartości elementów diagonalnych macierzy `Q` (parametr `dq(5)` programu) na wyniki regulacji.

   (d) Zbadać wpływ uwzględnienia czynnika zapewniającego zadany zapas stabilności układu na przebiegi regulacji.

Sprawozdanie

Sprawozdanie z przeprowadzonego laboratorium powinno zawierać:

- Wyznaczone macierze regulatorów i obserwatorów

- Wartości wskaźnika kwadratowego jakości
- Zawartość utworzonych podczas laboratorium skryptów MATLAB'a wraz z komentarzem
- Spostrzeżenia dotyczące doboru elementów macierzy R i Q (patrz wrór(2))
- Wykresy charakterystyki (patrz podpunkty (a)-(e) zadania 2) wraz z ich opisem.
- Uwagi i wnioski

Literatura:

1. http://www.engin.umich.edu/group/ctm/index.html
2. Gene F. Franklin, J. D. Powell , A. Emami-Naeini. - *Feedback Control of Dynamic Systems.* Prentice Hall, 4 edition, 2002
3. Kaczorek T. - *Teoria sterowania*, T.1, WNT Warszawa 1977.
4. Shahian B., Hassul M. -*Control System Design Using MALAB.* Prentice Hall, New Jersey,1993
5. *Control System Toolbox for Use with MATLAB.* User's Guide. MathWorks, 1992.

Skrypty Matlab'a

```
%%%%%%%%%%%%%% model wahadła odwóconego %%%%%%%%%%%%%%%%%%%%%%%%%%%%5

     M = 0.5;            %masa wózka
     m = 0.2;            %masa wahadła
     b = 0.1;            %współczynnik tarcia
     i = 0.006;
     g = 9.8;            %przyspieszenie ziemskie
     l = 0.3;            %długość wahadła

     p = i*(M+m)+M*m*l^2;                          %mianownik
     A = [0      1             0        0;
               0 -(i+m*l^2)*b/p (m^2*g*l^2)/p  0;
               0      0             0        1;
               0 -(m*l*b)/p    m*g*l*(M+m)/p 0];
     B = [0; (i+m*l^2)/p; 0; m*l/p];
     C = [1 0 0 0;
          0 0 1 0];
     D = [0;0];

%%%%%%%%%%%%%% funkcja rscale%%%%%%%%%%%%%%%%%%%

function[Nbar]=rscale(A,B,C,D,K)
% Dla danego układu liniowego:
%       .
%       x = Ax + Bu
%       y = Cx + Du
% i macierzy sterownika do stanu K,
%
% funkacja rscale(A,B,C,D,K) znaduje wspolczynnik N który wyeliminuje
% uchyb w stanie ustalonym wywołany sygnałem skokowym na wejściu (np. skok jednostkowy)
% korzystajac z nastepujacego schematu blokowego:
%
%                          /---------\
%      R         +     u  | .         |
%      ---> N --->() ---->| X=Ax+Bu |--> y=Cx ---> y
%               -|        \---------/
%                |             |
%                |<---- K <----|
%
%8/21/96 Yanjie Sun of the University of Michigan
%        under the supervision of Prof. D. Tilbury
%
s = size(A,1);
```

```
Z = [zeros([1,s]) 1];
N = inv([A,B;C,D])*Z'; Nx =
N(1:s);
Nu = N(1+s);
Nbar=Nu + K*Nx;

%%%% program lqopt.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nb=size(B); ni=nb(2); nc=size(C); no=nc(1);
r=1; dq=[1 10 100 1000 0.00001];     % elem. diag. macierzy R i Q
R=r*eye(ni);
t=linspace(0,10,500);
w=logspace(0,3,200);
clear hh; clear GG; clear pp; clear GGc; clear ppc;
for i=1:5,
  q=dq(i); Q=q*eye(size(A));
  Ae=A+0.4*eye(size(A));            % czynnik zwiększ. zapas stabilności
 %Ae=A;                              % bez zwiększ. zapasu stabilności
  [K,P]=lqr(Ae,B,Q,R);              % obliczenie wektora sprz. zwrotnego
  Co=K; Do=zeros(ni);               % macierze układu otwartego
  Ac=A-B*K; Bc=B; Cc=C; Dc=zeros(no,ni); % macierze układu zamkniętego
  h1=step(Ac,Bc,Cc,Dc,1,t);          % odp. skok. układu zamkniętego
  h=h1/h1(length(h1));               % normalizacja odpowiedzi skokowej
  [G,p]=bode(A,B,Co,Do,1,w);         % ch-ki czestotl. układu otwartego
  [Gc,pc]=bode(Ac,Bc,Cc,Dc,1,w);     % ch-ki czestotl. układu zamkniętego
  KK(i,:)=K;
  CLP(:,i)=eig(Ac);          % wart. własne macierzy układu zamkniętego
  hh(:,i)=h;
  GG(:,i)=G;
  pp(:,i)=p;
  GGc(:,i)=Gc;
  ppc(:,i)=pc;
end;
figure(1); plot(t,hh); title('Odpowiedzi skokowe układu zamkniętego');
           xlabel('t'); ylabel('h(t)');
figure(2);
subplot(2,1,1); loglog(w,GG); title('Układ otwarty'); xlabel('w');
        ylabel('G');
subplot(2,1,2); semilogx(w,pp); xlabel('w');
        ylabel('Faza');
figure(3);
subplot(2,1,1); loglog(w,GGc); title('Układ zamknięty');
xlabel('w'); ylabel('Gc');
subplot(2,1,2); semilogx(w,ppc);
xlabel('w'); ylabel('Faza');
```