Worksheet n°6: LQG control

The case of open-cavity flow

We consider the case of an open cavity at Re=6250, whose reduced-order model was obtained by the ERA algorithm for the stable subspace and an exact representation of the unstable subspace based on direct and adjoint global modes:

$$w(k+1) = Aw(k) + Bu(k)$$
$$y(k) = Cw(k)$$

Here (A, B, C) are matrices of dimension 14.

In NumStud6/ROM_6250_Control/, type:

Octave:

```
clear; close all;

readrom;

sys=ss(A,B,C,0,Ts);          % open-loop state-space system from u to y

figure(1);

subplot(2,1,1)

[mag,phase,ww]=bode(sys);

plot(ww(:),mag(:),'b');

max(abs(eig(A)))
```

a/ State-feedback

We look for a control law under the form $u(k) = Kw(k) + v(k)$, where v is the measurement noise. The closed-loop system is:

$$w(k+1) = Aw(k) + Bu(k) = (A + BK)w(k) + Bv(k)$$

$$y(k) = Cw(k)$$

The control gain K is defined by minimizing the closed-loop cost:

$$\mathfrak{I}(w,u) = \frac{1}{2}\sum_{k=0}^{\infty}(w(k)^* C^* C w(k) + l^2|u(k)|^2)$$

In NumStud6/ROM_6250_Control/, type:

Octave:

```
%%      Compute control gains
```

```
l2=1e10          % control cost (small-gain limit)

[X,CLeigs,KK]=dare(???)          % use dare to solve Riccati equation (to be completed)

K=???               % compute Control gain (to be completed)

max(abs(eig(A+B*K)))    % modulus of least-damped eigenvalue of state-feedback system

syssf=ss(A+B*K,B,[K;C],0,Ts)    % one input v, two outputs u and y

norm(syssf(2,1))           % performance in y

norm(syssf(1,1))           % performance in u
```

How do the performances in y and u evolve when $l^2 \to \infty$? when $l^2 \to 0$?

```
%%      Write control gain to file

fid=fopen('K.txt','wt');

fprintf(fid,'%21.14e\n',K(1:end));

fclose(fid);

 %%      State-feedback

figure(2)

impulse(syssf(2,1),'b') % impulse response of closed-loop system from input 1 (v) to output

                          % (1) y

%%      State-feedback in time-domain

gennoise;          % generate random noise signal for forced simulations in time domain

tim=load('time.txt');

v=load('noise.txt');

yy=lsim(syssf(2,1),v);         % generate output y from noise v with close-loop system

figure(3)

subplot(3,1,1)

plot(tim,v,'g');          % plot noise

subplot(3,1,2)

plot(tim,yy,'k');          % plot output y of closed-loop system

std(yy)                    % standard deviation of output

norm(syssf(2,1))              % norm of impulse response from input v to output y
```

b/ Observer

Introducing an estimated state for the control input ($u = Kw_e + v$), the governing equation reads:

$$w(k+1) = Aw(k) + BKw_e(k) + Bv(k), \qquad y(k) = Cw(k) + g(k)$$

The measurement y is corrupted by noise g. A dynamic observer may be obtained by determining L such that:

$$w_e(k+1) = Aw_e(k) + BKw_e(k) - L\big(y(k) - y_e(k)\big),$$
$$y(k) = Cw(k) + g(k), \qquad y_e(k) = Cw_e(k)$$

and such that $A + LC$ be stable.

The control gain L is defined by considering white-noise of variance Σ_v and Σ_g, $v = \sqrt{\Sigma_v}v'$ and $g = \sqrt{\Sigma_g}g'$. We would like to minimize the standard deviation of the estimation error $\sqrt{E(e^*e\,)}$, where the estimation error $e = w - w_e$ is driven by:

$$e(k+1) = w(k+1) - w_e(k+1) = (A+LC)e(k) + \begin{bmatrix}\sqrt{S_v}B & \sqrt{S_g}L\end{bmatrix}\begin{bmatrix}v'(k)\\ g'(k)\end{bmatrix}$$

The standard deviation of the estimation error $\sqrt{E(e^*e\,)}$ corresponds to the 2-norm of the impulse response of this system.

In NumStud6/ROM_6250_Control/, type:

Octave:

```
%%  Compute Kalman gains

Sv=1                    % actuator noise variance

Sg=1e10                 % measurement noise variance (small gain limit)

[Y,CLeigs,LL]=dare(???)             % use dare to solve Riccati equation (to be completed)

L=???                % compute Ricatti equation (to be completed)

max(abs(eig(A+L*C)))                % modulus of least-damped eigenvalues of error equation

syserr=ss(A+L*C,[B L],eye(size(A)),zeros(size(B,1),2),Ts);  % one input g, outputs w_1 to w_n

perfv=norm(syserr(1:end,1)) % error from v

perfg=norm(syserr(1:end,2)) % error from g
```

How does the standard deviation of the estimation error $\sqrt{E(e^*e\,)}$ (from v and g) evolve when $\Sigma_g/\Sigma_v \to \infty$? when $\Sigma_g/\Sigma_v \to 0$?

```
%%  write Kalman gains to file

fid=fopen('L.txt','wt');

fprintf(fid,'%21.14e\n',L(1:end));
```

```
fclose(fid);
```

To assess the properties of this dynamic observer, type:
Octave:

```
%%      Dynamic observer

sysdo=ss(A+B*K+L*C,-L,eye(size(A)),zeros(size(B)),Ts)   % one input y, outputs w_1 to w_n

figure(4)

impulse(sysdo(1,1),'b')     % impulse response of observer (first component)
```

c/ The compensator (combining the estimator and the controller) is given by:

$$w_e(k+1) = (A + BK + LC)w_e(k) - Ly(k)$$
$$u(k) = Kw_e(k) + v(k)$$

To assess the properties of the compensator, type in Octave:

```
%%      Compensator

sysc=ss(A+B*K+L*C,-L,K,0,Ts)% compensator, input y, output u

max(abs(eig(A+B*K+L*C)))    % modulus of least-damped eigenvalue of compensator

figure(1)

subplot(2,1,2)

[mag,phase,ww]=bode(sysc);

plot(ww(:),mag(:),'r');
```

d/ The closed-loop system is:

$$\begin{pmatrix} w(k+1) \\ w_e(k+1) \end{pmatrix} = \overbrace{\begin{pmatrix} A & BK \\ -LC & A+BK+LC \end{pmatrix}}^{A_{cl}} \begin{pmatrix} w(k) \\ w_e(k) \end{pmatrix} + \overbrace{\begin{pmatrix} B & 0 \\ 0 & -L \end{pmatrix}}^{B_{cl}} \begin{pmatrix} v(k) \\ g(k) \end{pmatrix},$$
$$\begin{pmatrix} u(k) \\ y(k) \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & K \\ C & 0 \end{pmatrix}}_{C_{cl}} \begin{pmatrix} w(k) \\ w_e(k) \end{pmatrix} + \begin{pmatrix} v(k) \\ g(k) \end{pmatrix}$$

The closed-loop numerical simulations (with large-scale matrices) are launched in NumStud6/ROM_6250_Control/ by typing:

```
FreeFem++ impulsedns_LQG.edp        %       impulse response in v with control

FreeFem++ noisedns_LQG.edp          %       noise in v with control
```

To analyze the properties of the closed-loop system, type in Octave:

```
%%     Closed-loop system

Acl=[[A B*K];[-L*C A+B*K+L*C]]    % closed-loop system

Bcl=[[B zeros(size(B))];[ zeros(size(B))  -L]]
```

```
Ccl=[[zeros(size(C)) K];[C zeros(size(C))]]

syscl=ss(Acl,Bcl,Ccl,eye(2,2),Ts)   % closed-loop system, two inputs v and g, two outputs
                                    % u  and y

max(abs(eig(Acl)))                  % modulus of least-damped eigenvalue of closed-loop
                                    % system

%%                                  % impulse response from v to y

figure(2)

hold on;

syscl21=syscl(2,1);

impulse(syscl21,'r');               % impulse response of closed-loop system

%%

norm(syssf(2,1))                    % performance with state-feedback

norm(syscl(2,1))                    % performance with observer-feedback

%%      Performance of closed-loop controller in time-domain with noise

figure(3)

subplot(3,1,3)

yy2=lsim(syscl(2,1),v);   % generate output y from noise v with close-loop system

plot(tim,yy2,'b');        % plot output y of closed-loop system

std(yy2)                  % standard deviation of output

norm(syscl(2,1))          % norm of impulse response from input 1 (v) to output 2 (y)

%%    performance of closed-loop controller in frequency domain with noise

 pkg load all

[spectra,freq]=pwelch(yy2,1000,0.5,1000,1/Ts);

% MATLAB  [spectra,freq]=pwelch(yy2,1000,500,1000,1/Ts);

figure(5)

subplot(3,1,1)

loglog(freq*2*pi,abs(spectra),'b')  % plot frequency spectrum of output y
```

```
%%      Compare frequency spectrum with predictions of rom
subplot(3,1,2)
[mag,phase,ww]=bode(syscl(2,1),freq*2*pi);
loglog(ww(:),mag(:),'b')    % plot frequency spectrum of output y
subplot(3,1,3)
[mag,phase,ww]=bode(syscl(1,1),freq*2*pi);
loglog(ww(:),mag(:),'r')    % plot frequency spectrum of output u
%%      Comparison of closed-loop impulse response between DNS and ROM
figure(6)
impulse(syscl(2,1),'r');                      % plot impulse response from v to y in ROM
% MATLAB : impulse(syscl(2,1)*Ts,'r');
hold on;
plotlinlin('impulse_LQG_Ts.txt',1,2,6);       % plot impulse response from v to y in DNS


 %%      Comparison of closed-loop response between DNS and ROM in presence of noise
figure(7)
ZZ=load('noise_LQG_Ts.txt');    % read output from DNS simulation
plot(tim(1:size(ZZ,1)),yy2(1:size(ZZ,1)),'b');  % plot output y of ROM with noise v
hold on;
plot(tim(1:size(ZZ,1)),ZZ(:,2),'r');    % plot output y of DNS with noise v
%%   Comparison of performances in ROM and DNS
std(yy2(1:size(ZZ,1)))          % standard deviation of output y in  ROM
std(ZZ(:,2))                    % standard deviation of output y in DNS
norm(syscl(2,1))                 % norm of impulse response from input 1 (v) to output 2 (y)
%%
```

e/ Stability robustness

Stability robustness properties are assessed by computing the gain margins g^+, g^-, ϕ^+.

In Octave, type:

```
%%      Stability robustness properties g+, g-, phi+

max(abs(zero(1-1*sys*sysc)))         % nominal system

max(abs(zero(1-3.46*sys*sysc)))      % gain margin 20 log10 g+ = 10.78dB

max(abs(zero(1-0.789*sys*sysc)))     % downside gain margin 20 log10 g- = -2.06dB

max(abs(zero(1-exp(i*0.328)*sys*sysc))) % phase margin phi+ = 18.8°
```

Compute the gain margins (g^+, g^-, ϕ^+) in:
- the medium-gain case: $l^2 = 10^2, \Sigma_g = 10^2, \Sigma_v = 1$
- the large-gain limit case: $l^2 = 10^{-2}, \Sigma_g = 10^{-2}, \Sigma_v = 1$

In which case do we observe the best stability-robustness properties?