Lecture 2: Exact identification

Ivan Markovsky

ELEC doctoral school 2013

Vrije Universiteit Brussel

Exercise 1: Check whether $w_\mathrm{d} \overset{?}{\in} \mathscr{B}$

- $w_\mathrm{d} = (u_\mathrm{d}, y_\mathrm{d}) = \big((0,1),(0,1),(0,1),(0,1)\big)$

```
w = [0 0 0 0; 1 1 1 1];
```

- $\mathscr{B} = \ker\big(R(\sigma)\big)$, where $R(z) = \begin{bmatrix}1 & -1\end{bmatrix} + \begin{bmatrix}-1 & 1\end{bmatrix} z$

```
R = [1 -1 -1 1]; ell = 1;
```

$w \overset{?}{\in} \ker\big(R(\sigma)\big)$

$$
\begin{aligned}
&\iff R(\sigma)w = 0 \\
&\iff R_0 w(t) + R_1 w(t+1) + \cdots + R_\ell w(t+\ell) = 0 \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for } t = 1, \ldots, T-\ell
\end{aligned}
$$

$$
\iff \underbrace{\begin{bmatrix} R_0 & R_1 & \cdots & R_\ell & & & \\ & R_0 & R_1 & \cdots & R_\ell & & \\ & & \ddots & \ddots & & \ddots & \\ & & & R_0 & R_1 & \cdots & R_\ell \end{bmatrix}}_{\mathscr{M}_T(R) \in \mathbb{R}^{\mathrm{p}(T-\ell)\times qT}} \underbrace{\begin{bmatrix} w(1) \\ w(2) \\ \vdots \\ w(T) \end{bmatrix}}_{\mathrm{vec}(w)} = 0
$$

$w \overset{?}{\in} \ker\big(R(\sigma)\big)$

$$
\begin{aligned}
\iff \quad & \mathscr{M}_T(R)\,\mathrm{vec}(w) = 0 \\
\iff \quad & R\,\mathscr{H}_{\ell+1}(w) = 0
\end{aligned}
$$

where

$$
\underbrace{\begin{bmatrix} R_0 & R_1 & \cdots & R_\ell \end{bmatrix}}_{R}
\underbrace{\begin{bmatrix}
w(1) & w(2) & \cdots & w(T-\ell) \\
w(2) & w(3) & \cdots & \\
\vdots & \vdots & & \vdots \\
w(\ell+1) & w(\ell+2) & \cdots & w(T)
\end{bmatrix}}_{\mathscr{H}_{\ell+1}(w) \in \mathbb{R}^{q(\ell+1)\times(T-\ell)}} = 0
$$

- compute $e = \|R\mathscr{H}_{\ell+1}(w)\|$ and check if $e < \varepsilon$

```
w = [0 0 0 0; 1 1 1 1];
R = [1 -1 -1 1]; ell = 1;
norm(R * blkhank(w, ell + 1))
```

```
   0
```

- `blkhank` constructs a block-Hankel matrix $\mathscr{H}_L(w)$

```
function H = blkhank(w, i, j)
[q, T] = size(w);
if T < q, w = w'; [q, T] = size(w); end
if nargin < 3, j = T - i + 1; end
H = zeros(i * q, j);
for ii = 1:i
  H(((ii - 1) * q + 1):(ii * q), :) ...
                 = w(:, ii:(ii + j - 1));
end
```

Homework

- use image representation to check

$$w \overset{?}{\in} \operatorname{image}\big(P(\sigma)\big)$$

- use state space representation to check

$$w \overset{?}{\in} \mathscr{B}(A, B, C, D)$$

Exercise 2: affine time-invariant system

- an LTI system $\mathscr{B} \in \mathscr{L}_{\mathtt{m},\ell}$ admits a kernel repr.

  $$\mathscr{B} = \ker\big(R(\sigma)\big) := \{w \mid R(\sigma)w = 0\}$$

  for some $R(z) = R_0 z^0 + R_1 z^1 + \cdots + R_\ell z^\ell$

- show that

  $$\mathscr{B}_c := \{\, w \mid R(\sigma)w = c \,\}$$

  is an affine time-invariant system, *i.e.*, $\mathscr{B}_c = \mathscr{B} + w_p$
  for LTI model $\mathscr{B} \in \mathscr{L}_{\mathtt{m},\ell}$ and trajectory w_p

- find $\mathscr{B}$ and w_p, s.t. $\mathscr{B} + w_p = \{\, w \mid (0.5 + \sigma)w = 1 \,\}$

- using the matrix representation of $R(\sigma)$

$$
\begin{aligned}
w \in \mathscr{B}_c \quad &\iff \quad \mathscr{M}_T(R)w = \mathbf{1}_{T-\ell} \otimes c =: \mathbf{c} \\
&\iff \quad \mathscr{M}_T(R)(w - w_\text{p}) = 0 \\
&\iff \quad w - w_\text{p} \in \ker\big(R(\sigma)\big) = \mathscr{B}
\end{aligned}
$$

- therefore, $\mathscr{B}_c = \mathscr{B} + w_\text{p}$, where $\mathscr{B} \in \mathscr{L}_{\mathtt{m},\ell}$ and

$$
\mathscr{M}_T(R)w_\text{p} = \mathbf{c}
$$

- *e.g.*, the least-norm solution

$$
w_\text{p} = \mathscr{M}_T^\top(R)\big(\mathscr{M}_T(R)\mathscr{M}_T^\top(R)\big)^{-1}\mathbf{c}
$$

- HW: find input/state/output representation of $\mathscr{B}_c$

- in the case of $\{w \mid (0.5+\sigma)w = 1\}$

```
T = 2; M = sylv([1/2 1], T);
wp = pinv(M) * ones(T, 1); disp(wp')
    0.4000    0.8000    0.4000    0.8000
```

- `sylv(R, T)` constructs the matrix $\mathscr{M}_T(R)$

```
function S = sylv(R, T)
nR = length(R); q = 2;
n = (nR / q) - 1;
S = zeros(T - n, q * T);
for i = 1:T - n
  S(i, (1:nR) + (i - 1) * q) = R;
end
```

Exercise 3: transfer function $\mapsto$ kernel repr.

- what model $\mathscr{B}_{\mathrm{tf}}(H)$ is specified by a transfer function

$$H(z) = \frac{q(z)}{p(z)} = \frac{q_0 + q_1 z^1 + \cdots + q_\ell z^\ell}{p_0 + p_1 z^1 + \cdots + p_\ell z^\ell}$$

- find R, such that

$$\mathscr{B}_{\mathrm{tf}}(H) = \ker(R)$$

- write a function `tf2ker` converting H (`tf` object) to R

- $H(z) = q(z)/p(z) \quad \overset{?}{\leftrightarrow} \quad R(z)$
- $y(z) = H(z)u(z) \quad \leftrightarrow \quad p(\sigma)y = q(\sigma)u$

$$\underbrace{\begin{bmatrix} q(\sigma) & -p(\sigma) \end{bmatrix}}_{R(\sigma)} \begin{bmatrix} u \\ y \end{bmatrix} = 0$$

- note: z may correspond to σ^{-1} as well as σ
- does $\mathscr{B}_{\text{tf}}(H)$ assume zero initial conditions?
  - if so,
$$\mathscr{B}_{\text{tf}}(H) = \{w \mid R(\sigma)(0 \wedge w) = 0\}$$
  - otherwise,
$$\mathscr{B}_{\text{tf}}(H) = \ker\big(R(\sigma)\big)$$

- note: MATLAB uses descending order of coefficients

```
function R = tf2ker(H)
[Q P] = tfdata(tf(H), 'v');
R = vec(fliplr([Q; -P]))';
```

Exercise 4: Specification of initial conditions

- initial conditions are explicitly specified in I/S/O repr.
- in MATLAB

```
LSIM(SYS,U,T,X0) specifies the initial
state vector X0 at time T(1)
  (for state-space models only).
```

- in transfer function representation initial conditions are often set to 0
- explain how to specify initial conditions in a representation free manner
- what is the link to $x_{\text{ini}} = x(1)$ in I/S/O repr?

- assuming that $\mathscr{B}$ is controllable
- initial conditions can be specified by prefix trajectory

$$w_{\text{ini}} = \big(w_{\text{ini}}(1), \dots, w_{\text{ini}}(T_{\text{ini}})\big)$$

i.e., by $w_{\text{ini}} \wedge w \in \mathscr{B}$

- the link between w_{ini} and x_{ini} is given by

$$y_{\text{ini}} = \mathscr{O}_\ell(A,C)A^{-\ell}x_{\text{ini}} + \mathscr{T}_\ell(A,B,C,D)u_{\text{ini}}$$

```
function x0 = inistate(w, sys)
l = size(sys, 'order');
x0 = obsv(sys) \ (w(1:l, 2) ...
                  - lsim(sys, w(1:l, 1)));
```

Exercise 5: Output matching

- given y_f and $\mathscr{B}$
- find u_f, such that $(u_\mathrm{f}, y_\mathrm{f}) \in \mathscr{B}$

Setup

- random SISO unstable system $\mathscr{B}$

```
clear all, n = 3;
Br = drss(n); [Qr, Pr] = tfdata(Br, 'v');
B = ss(tf(fliplr(Qr), fliplr(Pr), -1));
```

- reference output

```
T = 100; yf = ones(T, 1);
```

- $\mathscr{M}_T(R)w = 0 \quad \Longrightarrow \quad \mathscr{M}_T(P)u = \mathscr{M}_T(P)y$

```
R = tf2ker(B); M = sylv(R, T);
Mu = M(:, 1:2:end); My = - M(:, 2:2:end);
```

- many solutions (why?); compute a particular one

```
uf = pinv(Mu) * My * yf;
```

- $(u_\mathrm{f}, y_\mathrm{f}) \overset{?}{\in} \ker\big(R(\sigma)\big)$

```
disp(norm(R * blkhank([uf yf], n + 1)))
   1.1504e-14
```

- $(u_\mathrm{f}, y_\mathrm{f}) \overset{?}{\in} \operatorname{image}\big(P(\sigma)\big)$

```
disp(norm(M * vec([uf yf]')))
   1.1500e-14
```

- $(u_\mathrm{f}, y_\mathrm{f}) \overset{?}{\in} \mathscr{B}(A, B, C, D)$

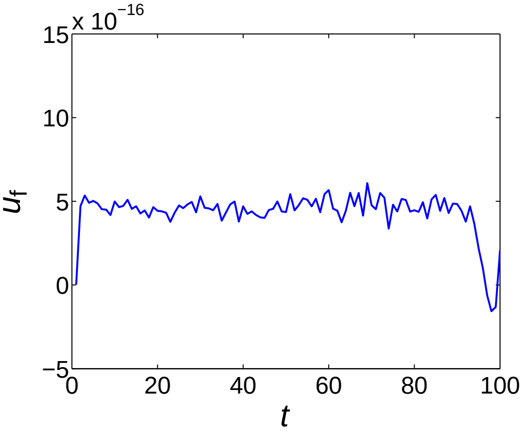
```
xini = inistate([uf yf], B);
disp(norm(yf - lsim(B, uf, [], xini)))
    0.7575
```

- where is the problem?

- the system is anti-stable
- the test $w \in \mathscr{B}(A,B,C,D)$ is ill-conditioned
- do backwards in time simulation

```
yfr = flipud(yf); ufr = flipud(uf);
xinir = inistate([ufr yfr], Br);
disp(norm(yfr - lsim(Br, ufr, [], xinir)))
   7.5809e-13
```

- particular (least squares) input



- HW: find all inputs