Initial conditions specification by trajectory

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

problem: given minimal $\mathscr{B} = \mathscr{B}(A, B, C, D) \in \mathscr{L}_{\mathrm{m},\ell}$

1. show that $\underbrace{\big(w(-\ell+1), \dots, w(0)\big)}_{w_{\mathrm{p}}} \in \mathscr{B}$ determines $x(0)$

2. explain how to use w_{p} to "set" given $x(0)$

3. implement and test $w_{\mathrm{p}} \leftrightarrow x(0)$ (`wp2x0` / `x02wp`)

Solution for part 1

$$y_{\mathrm{p}} = \underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{\ell-1} \end{bmatrix}}_{\mathscr{O}} x(-\ell+1) + \underbrace{\begin{bmatrix} H(0) & & & \\ H(1) & H(0) & & \\ \vdots & \ddots & \ddots & \\ H(\ell-1) & \cdots & H(1) & H(0) \end{bmatrix}}_{\mathscr{T}} u_{\mathrm{p}}$$

$$w_{\mathrm{p}} = \begin{bmatrix} u_{\mathrm{p}} \\ y_{\mathrm{p}} \end{bmatrix} \in \mathscr{B} \quad \Longrightarrow \quad \text{solution } x(-\ell+1) \text{ exists}$$

minimal repr. $\Longrightarrow$ $\mathscr{O}$ full rank $\Longrightarrow$ $x(-\ell+1)$ unique

$$x(0) = A^{\ell-1} x(-\ell+1) + \underbrace{\begin{bmatrix} A^{\ell-2}B & \cdots & BA^0 & 0 \end{bmatrix}}_{\mathscr{C}} u_{\mathrm{p}}$$

Solution for part 2 and 3

in order to set $x(0)$, we include a prefix $w_p \wedge w_f$

```
function x0 = wp2x0(wp, sys)
ell = size(sys, 'order');
xini  = obsv(sys) \ ...
        (wp(:, 2) - lsim(sys, wp(:, 1)));
C = [sys.b zeros(ell, 1)];
for i = 1:(ell - 2)
  C = [sys.a * C(:, 1) C];
end
x0 = sys.a ^ (ell - 1) * xini + C * wp(:, 1);
```

Solution for part 2 and 3

construct $\mathscr{C}$

```
C = [sys.b zeros(ell, 1)];
for i = 1:(ell - 2)
  C = [sys.a * C(:, 1) C];
end
```

construct $\mathscr{T}$

```
h = impulse(sys, ell - 1);
T = toeplitz(h, [h(1) zeros(1, ell - 1)]);
```

Solution for part 2 and 3 (continued)

$$x(0) = \begin{bmatrix} \mathscr{C} - A^{\ell-1}\mathscr{O}^{+}\mathscr{T} & A^{\ell-1}\mathscr{O}^{+} \end{bmatrix} \begin{bmatrix} u_{\mathrm{p}} \\ y_{\mathrm{p}} \end{bmatrix}$$

```
function wp = x02wp(x0, sys)
ell = size(sys, 'order');
<<construct-C>>
<<construct-T>>
O  = obsv(sys);
AO = sys.a ^ (ell -1) * pinv(O);
wp = pinv([C - AO * T , AO]) * x0;
wp = reshape(wp, ell, 2);
```

Solution (continued)

simulate data

```
n = 2; sys = drss(n);
T = 20; u = rand(T, 1); xini = rand(n, 1);
[y, t, x] = lsim(sys, u, [], xini); w = [u y];
```

test `wp2x0` and `x02wp`

```
<<simulate-data>>

wp = w(end - n + 1:end, :); x0 = x(end, :)';
wp2x0(wp, sys) - x0

wp2x0(x02wp(x0, sys), sys) - x0
```

Outline

Exact identification of a kernel representation

let $w \in \mathscr{B} \in \mathscr{L}^2_{1,\ell}$ (SISO system)

implement the method $w \mapsto R$ (slide 19)

test it on examples (use `drss`)

Solution

implementation

```
function r = w2r(w, ell)
r = null(blkhank(w, ell + 1)')';
```

test

```
<<simulate-data>>
sysh = r2tf(w2r(w, n));
norm(sys - sysh)
```

homework: generalize to the MIMO case

Impulse response estimation

let $w \in \mathscr{B} \in \mathscr{L}^2_{1,\ell}$ (SISO system)

implement the method $w \mapsto H$ (slide 20–21)

test it on examples (use `drss`)

Solution

implementation

```
function h = uy2h(u, y, ell, t)
L = ell + t;
H = [blkhank(u, L); blkhank(y, L)];
wini_uf = zeros(2 * ell + t, 1);
wini_uf(ell + 1) = 1;
h = H(2 * ell + t + 1:end, :) * ...
      pinv(H(1:(2 * ell + t), :)) * wini_uf;
```

test

```
<<simulate-data>>
t  = 5;
h  = impulse(sys, t - 1);
hh = uy2h(u, y, n, t);
norm(h - hh)
```

Outline

Misfit computation using image repr.

given

- data $w = \big(w(1), \dots, w(T)\big)$ and
- LTI system $\mathscr{B} = \text{image}\,\big(P(\sigma)\big)$

derive method for computing

$$\text{misfit}(w, \mathscr{B}) := \min_{\widehat{w} \in \mathscr{B}} \|w - \widehat{w}\|_2$$

i.e., find the orthogonal projection of w on $\mathscr{B}$

$w \overset{?}{\in} \text{image}\,(P(\sigma))$

$\iff$ there is v, such that $w = P(\sigma)v$

$\iff$ there is v, such that for $t = 1, \dots, T$

$$w(t) = P_0 v(t) + P_1 v(t+1) + \cdots + P_\ell v(t+\ell)$$

$\iff$ there is solution v of the system

$$\begin{bmatrix} w(1) \\ w(2) \\ \vdots \\ w(T) \end{bmatrix} = \underbrace{\begin{bmatrix} P_0 & P_1 & \cdots & P_\ell & & & \\ & P_0 & P_1 & \cdots & P_\ell & & \\ & & \ddots & \ddots & & \ddots & \\ & & & P_0 & P_1 & \cdots & P_\ell \end{bmatrix}}_{\mathscr{M}_{T+\ell}(P)} \begin{bmatrix} v(1) \\ v(2) \\ \vdots \\ v(T+\ell) \end{bmatrix}$$

Solution

we showed that

$$\widehat{w} \in \ker\big(R(\sigma)\big) \quad \iff \quad \widehat{w} = \mathscr{M}_T(P)v, \text{ for some } v$$

then the misfit computation problem

$$\text{misfit}(w, \mathscr{B}) := \min_{\widehat{w} \in \mathscr{B}} \|w - \widehat{w}\|$$

becomes

$$\text{minimize} \quad \text{over } v \quad \|w - \mathscr{M}_T(P)v\|$$

this is standard least-norm problem

projector on $\mathscr{B} = \text{image}(P)$

$$\Pi_{\text{image}(P)} := \mathscr{M}_T(P)\big(\mathscr{M}_T^\top(P)\mathscr{M}_T(P)\big)^{-1}\mathscr{M}_T^\top(P)$$

misfit

$$\text{misfit}(w,\mathscr{B}) := \sqrt{w^\top\big(I - \Pi_{\text{image}(P)}\big)w}$$

and optimal approximation

$$\widehat{w} = \Pi_{\text{image}(P)}w$$

homework: misfit computation with $\mathscr{B} = \ker\big(R(\sigma)\big)$

Misfit computation using I/S/O representation

given

- data $w = \big(w(1), \dots, w(T)\big)$ and
- LTI system $\mathscr{B} = \mathscr{B}(A, B, C, D)$

derive method for computing

$$\text{misfit}(w, \mathscr{B}) := \min_{\widehat{w} \in \mathscr{B}} \|w - \widehat{w}\|_2$$

i.e., find the orthogonal projection of w on $\mathscr{B}$

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

$$\mathscr{B}(A,B,C,D) = \{\, (u,y) \mid \sigma x = Ax + Bu,\ y = Cx + Du \,\}$$

$$(u_d, y_d) \in \mathscr{B}(A,B,C,D) \quad \iff \quad \exists\ x_{\text{ini}} \in \mathbb{R}^n, \text{ such that}$$

$$y = \underbrace{\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{T-1} \end{bmatrix}}_{\mathscr{O}_T(A,C)} x_{\text{ini}} + \begin{bmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & \ddots & \ddots & \ddots & \\ CA^{T-1}B & \cdots & CAB & CB & D \end{bmatrix} u$$

Solution

we showed that

$$\widehat{w} \in \mathscr{B}(A,B,C,D) \iff \widehat{y} = \mathscr{O}_T(A,C)\widehat{x}_{\text{ini}} + \mathscr{T}_T(H)\widehat{u}$$

then the misfit computation problem

$$\min_{\widehat{x}_{\text{ini}},\widehat{u}} \quad \left\| \begin{bmatrix} u_d \\ y_d \end{bmatrix} - \begin{bmatrix} 0 & I \\ \mathscr{O}_T(A,C) & \mathscr{T}_T(H) \end{bmatrix} \begin{bmatrix} \widehat{x}_{\text{ini}} \\ \widehat{u} \end{bmatrix} \right\|$$

exploiting the structure in the problem

$\rightsquigarrow$ EIV Kalman filter

Latency computation using kernel repr.

given

- data w and
- LTI system $\mathscr{B}_{\text{ext}} = \ker\big(R(\sigma)\big)$ $\qquad\qquad (w_{\text{ext}} := \left[\begin{smallmatrix}\widehat{e} \\ w\end{smallmatrix}\right])$

find an algorithm for computing

$$\text{minimize} \quad \text{over } e \quad \|\widehat{e}\| \quad \text{subject to} \quad (\widehat{e}, w) \in \mathscr{B}_{\text{ext}}$$

Solution

partition $R = \begin{bmatrix} R_e & R_w \end{bmatrix}$ conformably with $w_{\text{ext}} = \left[\begin{smallmatrix} e \\ w \end{smallmatrix}\right]$

by analogy with the derivation on page 41, we have

$$\begin{bmatrix} e \\ w \end{bmatrix} \in \ker\big(R(\sigma)\big) \iff \begin{bmatrix} \mathscr{M}_T(R_e) & \mathscr{M}_T(R_w) \end{bmatrix} \begin{bmatrix} e \\ w \end{bmatrix} = 0$$

the latency computation problem is

$$\min_{e} \quad \|e\|_2 \quad \text{subject to} \quad \mathscr{M}_T(R_e)e = -\mathscr{M}_T(R_w)w$$

the solution is given by

$$\widehat{e} = -\underbrace{\big(\mathscr{M}_T(R_e)^\top \mathscr{M}_T(R_e)\big)^{-1} \mathscr{M}_T(R_e)^\top}_{\mathscr{M}_T(R_e)^+} \mathscr{M}_T(R_w)w$$

Outline

Software

mosaic-Hankel low-rank approximation

http://slra.github.io/software.html

```
[sysh,info,wh] = ident(w, m, ell, opt)
```

- `sysh` — I/S/O representation of the identified model
- `opt.sys0` — I/S/O repr. of initial approximation
- `opt.wini` — initial conditions
- `opt.exct` — exact variables
- `info.Rh` — parameter *R* of kernel repr.
- `info.M` — misfit

```
[M, wh, xini] = misfit(w, sysh, opt)
```

demo file

Variable permutation

verify that permutation of the variables doesn't change the optimal misfit

```
T = 100; n = 2; B0 = drss(n);
u = randn(T, 1); y = lsim(B0, u) + 0.001 * ranc
[B1, info1] = ident([u y], 1, n); disp(info1.M)
   2.9736e-05
[B2, info2] = ident([y u], 1, n); disp(info2.M)
   2.9736e-05
disp(norm(B1 - inv(B2)))
   5.8438e-12
```

Output error identification

verify that the results of `oe` and `ident` coincide

```
T = 100; n = 2; B0 = drss(n);
u = randn(T, 1); y = lsim(B0, u) + 0.001 * ranc
opt = oeOptions('InitialCondition', 'estimate')
B1 = oe(iddata(y, u), [n + 1 n 0], opt);
B2 = ident([u y], 1, n, struct('exct', 1));
norm(B1 - B2) / norm(B1)

ans =

   1.4760e-07
```