The Behavioral Toolbox

Ivan Markovsky

ICREA

CIMNE
INTERNATIONAL CENTRE
FOR NUMERICAL METHODS
IN ENGINEERING

How to check if two LTI systems are equal?

we would like this MATLAB code to give ’true’

```
m = 2; p = 2; n = 3;
sys1 = drss(n, p, m);
sys2 = ss2ss(sys1, rand(n));
sys1 == sys2
```

however, it gives an error:

```
Operator '==' is not supported
for operands of type 'ss'.
```

why are `ss` objects not comparable?

A system is a set of signals (the behavior)

signals

$w \in (\mathbb{R}^q)^{\mathbb{N}}$, $w : \mathbb{N} \to \mathbb{R}^q$	q-variate discrete-time signal
$w\vert_T := \big(w(1), \dots, w(T)\big)$	restriction of w to $[1, T]$
σ, $(\sigma w)(t) := w(t+1)$	unit shift operator

systems

$\mathscr{B} \subset (\mathbb{R}^q)^{\mathbb{N}}$	q-variables discrete-time system
$\mathscr{B}\vert_T := \{\, w\vert_T \mid w \in \mathscr{B} \,\}$	restriction of $\mathscr{B}$ to $[1, T]$
$\mathscr{L}^q_{(m,\ell,n)}$	bounded complexity LTI systems

$\mathbf{m}(\mathscr{B})$ / $\boldsymbol{\ell}(\mathscr{B})$ / $\mathbf{n}(\mathscr{B})$ — # of inputs / lag / order

$\mathscr{B}$ is represented by basis a B_T for $\mathscr{B}|_T$

$$\mathscr{B}|_T = \text{image} \begin{bmatrix} b^1 & \cdots & b^r \end{bmatrix} \qquad (B_T)$$

(B_T) is nonparameteric representation of $\mathscr{B}|_T$

$$w \in \mathscr{B}|_T \iff w = B_T g, \; g \in \mathbb{R}^r$$

$\mathscr{B} \in \mathscr{L}_c^q$ and $T \geq \boldsymbol{\ell}(\mathscr{B})$ implies that

$$r = \dim \mathscr{B}|_T = \mathbf{m}(\mathscr{B})T + \mathbf{n}(\mathscr{B}) \qquad (\dim \mathscr{B}|_T)$$

$\mathscr{B}|_{\boldsymbol{\ell}(\mathscr{B})+1}$ defines $\mathscr{B}$ $\implies$ (B_T) is representation of $\mathscr{B}$

B_T is obtained from data or representations

$$\text{data: } \mathscr{W}_\text{d} = \{\, w_\text{d}^1, \dots, w_\text{d}^N \,\} \mapsto B_T \qquad (\texttt{w2BT})$$

$$(A, B, C, D) \mapsto B_T = \Pi \begin{bmatrix} 0_{mT \times n} & I_{mT} \\ \mathscr{O}_T(A,C) & \mathscr{T}_T(H) \end{bmatrix} \qquad (\texttt{ss2BT})$$

$$\text{kernel: } R \mapsto B_T := \begin{bmatrix} b^1 & \cdots & b^r \end{bmatrix} \qquad (\texttt{R2BT})$$

indirect “data-driven approach” $(\texttt{B2BT})$

1. simulate data $\mathscr{W}_\text{d}$ from parametric representation
2. use `w2BT` to obtain (B_T) from the data

B_T is structured, due to the LTI dynamics

the structure is fully revealed when $T \geq \boldsymbol{\ell}(\mathscr{B})$

$\mathscr{B}|_T$ is $(mT+n)$-dimensional shift-invariant subspace

complexity bounded $\iff$ $\dim \mathscr{B}|_T < qT$

$$\mathscr{B} \in \mathscr{L}^q_{(m,\ell,n)} \implies \dim \mathscr{B}|_T = mT+n = \operatorname{rank} B_T$$

time-invariance $\iff$ shift-invariance

- autonomous case: sum-of-exponentials
- open systems: the structure is hidden
- image $\mathscr{H}_T(w_\mathrm{d})$ imposes shift-invariance by construction

Computing B_T has hidden dangers

it involves rank computation

$$\dim \mathscr{B}|_T = qT - \operatorname{rank} \mathscr{M}_T(R) = \operatorname{rank} \mathscr{H}_T(\mathscr{W}_\mathrm{d})$$

rank is computed by singular values thresholding

$$\widehat{r} := \#\text{ of singular values} \geq \texttt{tol} \qquad (\widehat{r})$$

the data has to be informative

$$\operatorname{rank} \mathscr{H}_T(\mathscr{W}_\mathrm{d}) = \mathbf{m}(\mathscr{B})T + \mathbf{n}(\mathscr{B}) \qquad \text{(GPE)}$$

The SVD approximation doesn't impose time-invariant structure on B_T

two options for complexity estimation:

1. specify `tol`, in which case r is found from ($\widehat{r}$), or
2. $c := (m, \ell, n)$, in which case r is found from ($\dim \mathscr{B}|_T$)

SVD approximation

when "small" singular values are discarded,
B_T is not shift-invariant and therefore
does not represent $\mathscr{B}|_T$ of $\mathscr{B} \in \mathscr{L}_c^q$

Back to the example of systems equality

equality in the behavioral setting: $\mathscr{B}^1 \stackrel{?}{=} \mathscr{B}^2$

using (B_T) with $T \geq \max\{\boldsymbol{\ell}(\mathscr{B}^1), \boldsymbol{\ell}(\mathscr{B}^2)\}$

$$\mathscr{B}^1 = \mathscr{B}^2 \quad \iff \quad \mathscr{B}^1|_T = \mathscr{B}^2|_T$$

possible implementation

```
T = max(lag(B1), lag(B2));
BT1 = B2BT(B1, T);
BT2 = B2BT(B2, T);
rank([BT1 BT2]) == rank(BT1)
```

analysis

- complexity computation `BT2c`
- input/output partitioning `is_io`, `BT2IO`
- controllability test `isunctr`, `distunctr`

parametric representations

- kernel `BT2R`, `R2BT`
- input/state/output `BT2ss`, `ss2BT`
- conversions `R2ss`, `R2Rmin`

identification and signal-processing

- exact identification `w2BT`, `w2R`, `w2ss`
- approximate identification `ident`
- data-driven signal-processing `ddint`