System theory without state-space and transfer functions? Yes, it's possible!

Ivan Markovsky

VRIJE UNIVERSITEIT BRUSSEL

The distinction between model-based and model-free depends on the notion of model

the classical notion is narrow and fragmented

- convolution model (non-parametric)
- state-space model (parametric)
- transfer function model (can be either)

alternative behavioral notion: set of trajectories

- if any model is allowed, constraints are not imposed
- any method is model-based in the behavioral sense

the key feature of model is complexity restriction

$\mathscr{B}_1$ less complex than $\mathscr{B}_2 \quad \Longleftrightarrow \quad \mathscr{B}_1 \subset \mathscr{B}_2$

in the LTI case, complexity $\leftrightarrow$ dimension

complexity: (# inputs, order, lag)

$$\mathbf{c}(\mathscr{B}) := \big(\mathbf{m}(\mathscr{B}), \mathbf{n}(\mathscr{B}), \mathbf{l}(\mathscr{B})\big)$$

$\mathscr{L}_c$ — bounded complexity LTI model class

Data-driven representation (exact data)

restriction of w and $\mathscr{B}$ to finite interval $[1, L]$

$$w|_L := \big(w(1), \dots, w(L)\big), \quad \mathscr{B}|_L := \{\, w|_L \mid w \in \mathscr{B} \,\}$$

for $w_\mathrm{d} = \big(w_\mathrm{d}(1), \dots, w_\mathrm{d}(T)\big)$ and $1 \leq L \leq T$

$$\mathscr{H}_L(w_\mathrm{d}) := \begin{bmatrix} (\sigma^0 w_\mathrm{d})|_L & (\sigma^1 w_\mathrm{d})|_L & \cdots & (\sigma^{T-L} w_\mathrm{d})|_L \end{bmatrix}$$

$$(\sigma w)(t) := w(t+1), \quad \sigma\mathscr{B} := \big\{\sigma w \mid w \in \mathscr{B}\big\}$$

define $\quad \widehat{\mathscr{B}}|_L := \operatorname{image} \mathscr{H}_L(w_\mathrm{d})$

Conditions for informativity of the data

$\mathscr{B}|_L = \text{image}\, \mathscr{H}_L(w_\text{d})$ if and only if

$$\text{rank}\, \mathscr{H}_L(w_\text{d}) = L\mathbf{m}(\mathscr{B}) + \mathbf{n}(\mathscr{B})$$

I. Markovsky and F. Dörfler, Identifiability in the Behavioral Setting, 2020
https://imarkovs.github.io/publications/identifiability.pdf

sufficient conditions (“fundamental lemma”):

1. $w_\text{d} = \begin{bmatrix} u_\text{d} \\ y_\text{d} \end{bmatrix}$
2. $\mathscr{B}$ controllable
3. $\mathscr{H}_{L+\mathbf{n}(\mathscr{B})}(u_\text{d})$ full row rank

J.C. Willems et al., A note on persistency of excitation
Systems & Control Letters, (54)325–329, 2005

Generic data-driven problem: trajectory interpolation/approximation

given: "data" trajectory $w_{\rm d} \in \mathscr{B}|_T$

partially specified trajectory $w|_{I_{\rm given}}$

($w|_{I_{\rm given}}$ selects the elements of w, specified by $I_{\rm given}$)

aim:

$$\text{minimize} \quad \text{over } \widehat{w} \quad \| w|_{I_{\rm given}} - \widehat{w}|_{I_{\rm given}} \| \quad \text{subject to} \quad \widehat{w} \in \mathscr{B}|_L$$

solution:

$$\widehat{w} = \mathscr{H}_L(w_{\rm d}) \big(\mathscr{H}_L(w_{\rm d})|_{I_{\rm given}} \big)^+ w|_{I_{\rm given}}$$

Special cases

simulation

- given data: initial conditions and input
- to-be-found: output (exact interpolation)

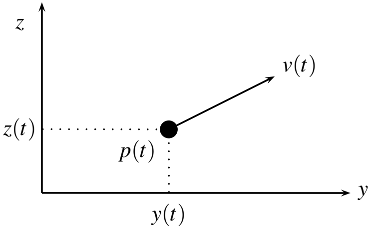
smoothing

- given data: noisy trajectory
- to-be-found: ℓ_2-optimal approximation

tracking

- given data: to-be-tracked trajectory
- to-be-found: ℓ_2-optimal approximation

Example: predicting free fall trajectory without knowing the laws of physics



p — position
v — velocity

$p(0), v(0)$ — initial conditions

goal: given initial conditions, find the trajectory p

- model-based approach:
  1. physics $\mapsto$ model
  2. model + ini. cond. $\mapsto p$
- data-driven approach: data $p_d^1, \ldots, p_d^N$ + ini. cond. $\mapsto p$

Data-driven free fall prediction method assuming that the data $p_d^1, \ldots, p_d^N$ is exact

informative data condition:

$$\text{rank} \underbrace{\begin{bmatrix} p_d^1 & \cdots & p_d^N \end{bmatrix}}_{P_d} = 5$$

algorithm:

1. solve $\begin{bmatrix} p_d^1(0\delta) & \cdots & p_d^N(0\delta) \\ p_d^1(1\delta) & \cdots & p_d^N(1\delta) \\ p_d^1(2\delta) & \cdots & p_d^N(2\delta) \end{bmatrix} g = \underbrace{\begin{bmatrix} p(0\delta) \\ p(1\delta) \\ p(2\delta) \end{bmatrix}}_{\text{ini. cond.}}$ $\left(\delta \text{ — sampling period}\right)$
2. define $p := P_d g$

Generalizations

multiple data trajectories $w_{\text{d}}^1, \dots, w_{\text{d}}^N$

$$\mathscr{B} = \text{image} \begin{bmatrix} \mathscr{H}_L(w_{\text{d}}^1) & \cdots & \mathscr{H}_L(w_{\text{d}}^N) \end{bmatrix}$$

w_{d} not exact / noisy

maximum-likelihood estimation
$\rightsquigarrow$ Hankel structured low-rank approximation/completion
nuclear norm and ℓ_1-norm relaxations
$\rightsquigarrow$ nonparametric, convex optimization problems

nonlinear systems

results for special classes of nonlinear systems:
Volterra, Wiener-Hammerstein, bilinear, ...

Two ways of doing complexity reduction: constrained optimization and regularization

1. imposing the hard constraint: $\mathscr{B} \in \mathscr{L}_c$
   - the complexity c is fixed and given
   - requires parametric representation
   - $\mathscr{L}_c$ has a manifold structure $\rightsquigarrow$ nonconvex optimization

2. using soft constraints (*e.g.*, $+\lambda \|g\|_1$ term)
   - uses non-parametric representation (*e.g.*, $w = \mathscr{H}_L(w_\text{d})g$)
   - requires tuning of a hyper-parameter, *e.g.*,

$$\lambda \uparrow \implies \text{sparser } g \implies \text{simpler model}$$

   - can be used for other types of priori knowledge