Behavioral Approach to Data-Driven System Theory and Control

Ivan Markovsky

VRIJE UNIVERSITEIT BRUSSEL

Take-home messages

bothersome inconsistencies lead to new ideas

useful ideas lead to algorithms

the ℓ_1-norm heuristic is (unreasonably) effective

Outline

Classical vs behavioral approaches

Data-driven interpolation and approximation

Convex relaxations and empirical validation

Outline

Classical vs behavioral approaches

Data-driven interpolation and approximation

Convex relaxations and empirical validation

The classical approach views system as input-output map

$$\text{input} \longrightarrow \boxed{\text{system}} \longrightarrow \text{output}$$

the system is a signal processor

accepts input and produces output signal

intuition: the input causes the output

The input-output map view of the system is deficient: it ignores the initial condition

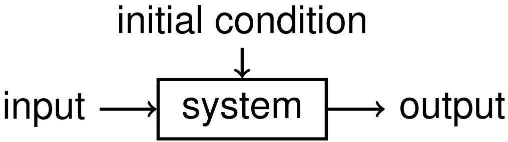
example: mass driven by external force

- input $\leftrightarrow$ force
- output $\leftrightarrow$ position
- ??? $\leftrightarrow$ position and velocity at start (initial condition)

input-output maps assume zero initial condition

how to account for nonzero initial condition?

Taking into account the initial condition leads to the state-space approach



paradigm shift from "classical" to "modern"

classical: scalar transfer function

modern: multivariable state-space

The modern state-space paradigm brought new theory, problems, and methods

state-space theory

- manifests the "finite memory" structure of the system
- brought the concepts of controllability and observability
- deals seamlessly with time-varying and MIMO systems

new problems / solution methods

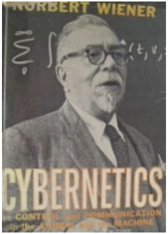
- linear quadratic optimal control (LQ control)
- optimal state estimation (the Kalman filter)
- balanced model reduction

amenable for numerical computations

A case in point: optimal filtering (signal from noise separation)

Wiener filter (1942)

- transfer functions approach
- assumes stationarity
- no practical real-time method



Kalman filter (1960)

- state-space approach
- non-stationary processes
- recursive real-time solution

There are other awkward things with the input/output thinking

modeling from first principles leads to relations

the input/output partitioning is not unique

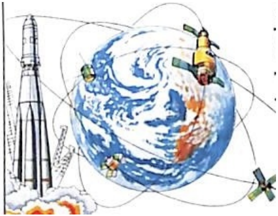
interconnection of systems is variables sharing

First principles modeling leads to relations

natural phenomena rarely operate as signal processors

the variables of interest satisfy relations, not functions

example: planetary orbits



More basic example: Ohmic resistor voltage and current satisfy relation

to-be-modeled variables: voltage V and current I

Ohm's law:

- $V = RI$, with R the resistance
- $I = CV$, with $C := 1/R$ the conductance

Q: how to fit the limit cases

- open circuit — $R = \infty$, $C = 0$
- short circuit — $R = 0$, $C = \infty$

neatly in a unified framework?

A: V, I satisfy (linear) relation

The behavioral approach was born from a critical revision of the input/output thinking

simple idea: the system is set of trajectories

- variables not partitioned into inputs and outputs
- the system is separated from its representations

the input/output approach is a special case

relevant for the emerging data-driven methods

The behavior is all that matters

"The operations allowed to bring model equations in a more convenient form are exactly those that do not change the behavior. Dynamic modeling and system identification aim at coming up with a specification of the behavior. Control comes down to restricting the behavior."

Jan C. Willems (1939–2013)

J. C. Willems, "The behavioral approach to open and interconnected systems: Modeling by tearing, zooming, and linking," Control Systems Magazine, vol. 27, pp. 46–99, 2007.

Analogy with solution of systems of equations

Q: what operations are allowed?

A: the ones that don't change the solution set

(for linear systems, the "elementary operations")

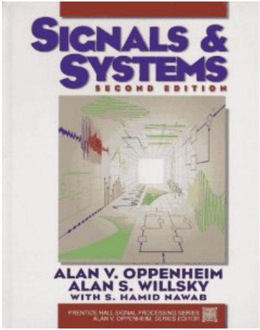
the solution set is all that matters

Classical definition of linear system

$S : u \mapsto y$ is linear $\iff$ S is linear function

for all u, v and $\alpha, \beta \in \mathbb{R}$,

$$S : \alpha u + \beta v \mapsto \alpha S(u) + \beta S(v)$$



The classical definition is deficient

(silently) assumes

- zero initial condition
- controllability

doesn't apply to autonomous systems

relaxing the assumptions requires state-space

Behavioral definition of linear system

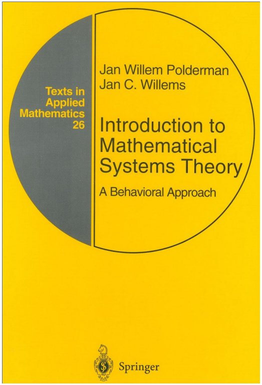
$\mathscr{B}$ is linear $\iff$ $\mathscr{B}$ is subspace

for all $w, v \in \mathscr{B}$ and $\alpha, \beta \in \mathbb{R}$

$$\alpha w + \beta v \in \mathscr{B}$$

fixes the issues with

- nonzero initial condition
- autonomous systems
- controllable systems



Summary: behavioral approach

detach the system from its representations

- define properties and problems in terms of the behavior
- lead to new, more general, definitions and problems
- avoid inconsistencies of the classical approach

separate problem from solution methods

- different representations lead to different methods
- show links among different methods
- lead to new solutions

naturally suited for the "data-driven paradigm"

Paradigms shifts

1940–1960	classical	SISO transfer function
1960–1980	modern	MIMO state-space
1980–2000	behavioral	the system as a set
2000–now	data-driven	using directly the data

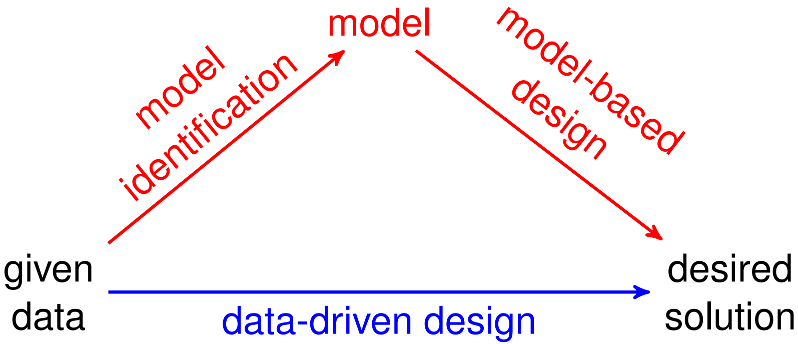
Outline

Classical vs behavioral approaches

Data-driven interpolation and approximation

Convex relaxations and empirical validation

The new "data-driven" paradigm obtains desired solution directly from given data



Data-driven does not mean model-free

data-driven problems do assume model

however, specific representation is not fixed

the methods we review are non-parametric

A dynamical system $\mathscr{B}$ is a set of signals

$$\begin{aligned} w \in \mathscr{B} \quad & \leftrightarrow \quad \text{"}w \text{ is trajectory of } \mathscr{B}\text{"} \\ & \leftrightarrow \quad \text{"}\mathscr{B} \text{ is exact model for } w\text{"} \end{aligned}$$

$\mathscr{B}$ is linear system $\quad :\Longleftrightarrow \quad$ $\mathscr{B}$ is subspace

$\mathscr{B}$ is time-invariant $\quad :\Longleftrightarrow \quad \sigma\mathscr{B} = \mathscr{B}$

$(\sigma w)(t) := w(t+1)$ — shift operator

$$\sigma\mathscr{B} := \{\sigma w \mid w \in \mathscr{B}\}$$

"good definition should formalize sensible intuition"

The set of linear time-invariant systems $\mathscr{L}$ has structure characterized by set of integers

the dimension of $\mathscr{B} \in \mathscr{L}$ is determined by

$\mathbf{m}(\mathscr{B})$ — number of inputs

$\mathbf{n}(\mathscr{B})$ — order ($=$ minimal state dimension)

$\mathbf{l}(\mathscr{B})$ — lag ($=$ observability index)

J.C. Willems, From time series to linear systems.
Part I, Finite dimensional linear time invariant systems, Automatica, 22(561–580), 1986

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

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 (infinite horizon)

data: exact infinite trajectory w_{d} of $\mathscr{B} \in \mathscr{L}$

define $\quad \widehat{\mathscr{B}} := \mathrm{span}\{\, w_{\mathrm{d}}, \sigma w_{\mathrm{d}}, \sigma^2 w_{\mathrm{d}}, \dots \}$

identifiability condition: $\quad \mathscr{B} = \widehat{\mathscr{B}}$

Data-driven representation (finite horizon)

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 $1 \le L \le T$

$$\mathscr{H}_L(w_\mathrm{d}) := \Big[(\sigma^0 w_\mathrm{d})|_L \quad (\sigma^1 w_\mathrm{d})|_L \quad \cdots \quad (\sigma^{T-L} w_\mathrm{d})|_L \Big]$$

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

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

I. Markovsky and F. Dörfler, Identifiability in the Behavioral Setting, 2020

sufficient conditions (input design perspective):

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 (PE)

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

PE — persistency of excitation, GPE — generalized PE

Generic data-driven problem: trajectory interpolation/approximation

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

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

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

aim: minimize over $\widehat{w}$ $\quad \| w|_{I_\text{given}} - \widehat{w}|_{I_\text{given}} \|$

subject to $\widehat{w} \in \mathscr{B}|_L$

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

Special cases

simulation

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

smoothing

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

tracking control

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

Generalizations

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

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

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, ...

Summary: data-driven signal processing

data-driven representation

leads to general, simple, practical methods

interpolation/approximation of trajectories

simulation, filtering and control are special cases
assumes only LTI dynamics; no hyper parameters

dealing with noise and nonlinearities

nonlinear optimization
convex relaxations

Outline

Classical vs behavioral approaches

Data-driven interpolation and approximation

Convex relaxations and empirical validation

The data w_d being exact vs inexact / "noisy"

w_d exact and satisfying (GPE)

- "system theory" problems
- image $\mathscr{H}_L(w_\mathrm{d})$ is nonparametric finite-horizon model
- data-driven solution $=$ model-based solution

w_d inexact, due to noise and/or nonlinearities

- naive approach: apply the solution (SOL) for exact data
- rigorous: assume noise model $\rightsquigarrow$ ML estimation problem
- heuristics: convex relaxations of the ML estimator

The maximum-likelihood estimation problem in the errors-in-variables setup is nonconvex

errors-in-variables setup: $w_\mathrm{d} = \overline{w}_\mathrm{d} + \widetilde{w}_\mathrm{d}$

- $\overline{w}_\mathrm{d}$ — true data, $\overline{w}_\mathrm{d} \in \mathscr{B}|_T$, $\mathscr{B} \in \mathscr{L}_c^q$
- $\widetilde{w}_\mathrm{d}$ — zero mean, white, Gaussian measurement noise

ML problem: given w_d, c, and $w|_{I_\mathrm{given}}$

$$\begin{aligned}
\underset{g}{\text{minimize}} \quad & \| w|_{I_\mathrm{given}} - \mathscr{H}_L(\widehat{w}_\mathrm{d}^*)|_{I_\mathrm{given}} g \| \\
\text{subject to} \quad & \widehat{w}_\mathrm{d}^* = \arg\min_{\widehat{w}_\mathrm{d}, \widehat{\mathscr{B}}} \quad \| w_\mathrm{d} - \widehat{w}_\mathrm{d} \| \\
& \qquad \text{subject to} \quad \widehat{w}_\mathrm{d} \in \widehat{\mathscr{B}}|_T \text{ and } \widehat{\mathscr{B}} \in \mathscr{L}_c^q
\end{aligned}$$

The ML estimation problem is equivalent to Hankel structured low-rank approximation

$$\begin{aligned}
\underset{g}{\text{minimize}} \quad & \| w|_{I_\text{given}} - \mathscr{H}_L(\widehat{w}_\text{d}^*)|_{I_\text{given}} g \| \\
\text{subject to} \quad & \widehat{w}_\text{d}^* = \arg\min_{\widehat{w}_\text{d}, \widehat{\mathscr{B}}} \quad \| w_\text{d} - \widehat{w}_\text{d} \| \\
& \qquad \text{subject to} \quad \widehat{w}_\text{d} \in \widehat{\mathscr{B}}|_T \text{ and } \widehat{\mathscr{B}} \in \mathscr{L}_c^q
\end{aligned}$$

$$\Updownarrow$$

$$\begin{aligned}
\underset{g}{\text{minimize}} \quad & \| w|_{I_\text{given}} - \mathscr{H}_L(\widehat{w}_\text{d}^*)|_{I_\text{given}} g \| \\
\text{subject to} \quad & \widehat{w}_\text{d}^* = \arg\min_{\widehat{w}_\text{d}} \quad \| w_\text{d} - \widehat{w}_\text{d} \| \\
& \qquad \text{subject to} \quad \operatorname{rank} \mathscr{H}_{\ell+1}(\widehat{w}_\text{d}) \leq (\ell+1)m + n
\end{aligned}$$

Solution methods

local optimization

- choose a parametric representation of $\widehat{\mathscr{B}}(\theta)$
- optimize over $\widehat{w}$, $\widehat{w_\text{d}}$, and θ
- depends on the initial guess

convex relaxation based on the nuclear norm

$$\text{minimize} \quad \text{over } \widehat{w}_\text{d} \text{ and } \widehat{w} \quad \|w|_{I_\text{given}} - \widehat{w}|_{I_\text{given}}\| + \|w_\text{d} - \widehat{w}_\text{d}\| + \gamma \cdot \left\| \begin{bmatrix} \mathscr{H}_\Delta(\widehat{w}_\text{d}) & \mathscr{H}_\Delta(\widehat{w}) \end{bmatrix} \right\|_*$$

convex relaxation based on ℓ_1-norm (LASSO)

$$\text{minimize} \quad \text{over } g \quad \|w|_{I_\text{given}} - \mathscr{H}_L(w_\text{d})|_{I_\text{given}} g\| + \lambda \|g\|_1$$

Empirical validation on real-life datasets

	data set name	T	m	p
1	Air passengers data	144	0	1
2	Distillation column	90	5	3
3	pH process	2001	2	1
4	Hair dryer	1000	1	1
5	Heat flow density	1680	2	1
6	Heating system	801	1	1

G. Box, and G. Jenkins. Time Series Analysis: Forecasting and Control, Holden-Day, 1976

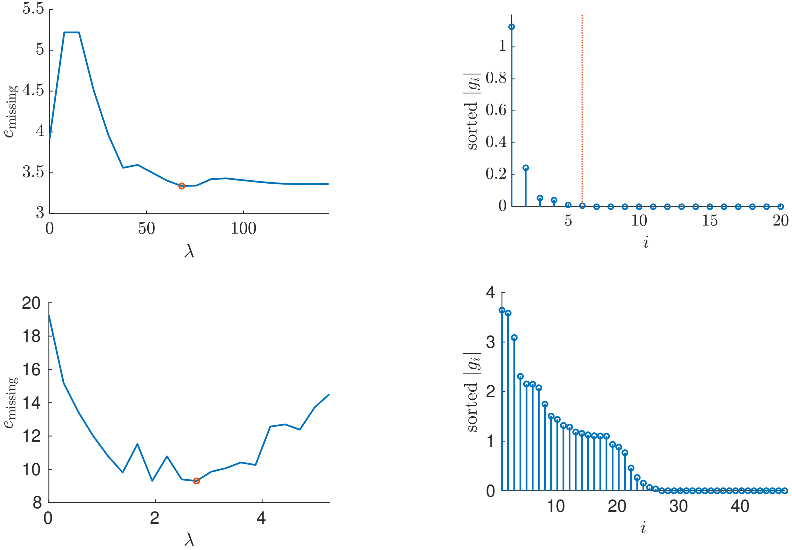
B. De Moor, et al.*DAISY: A database for identification of systems. Journal A, 38:4–5, 1997*

ℓ_1-norm regularization with optimized λ achieves the best performance

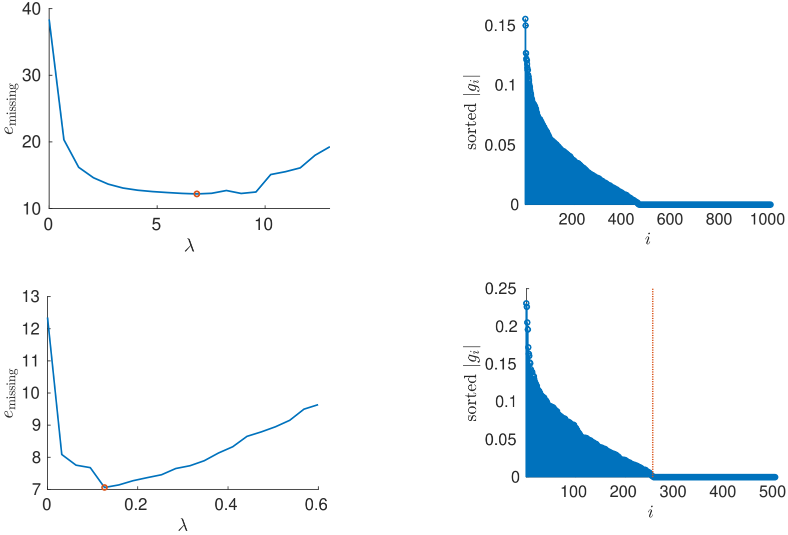
$$e_{\text{missing}} := \frac{\| w|_{I_{\text{missing}}} - \widehat{w}|_{I_{\text{missing}}} \|}{\| w|_{I_{\text{missing}}} \|} \, 100\%$$

	data set name	naive	ML	LASSO
1	Air passengers data	3.9	fail	3.3
2	Distillation column	19.24	17.44	9.30
3	pH process	38.38	85.71	12.19
4	Hair dryer	12.35	8.96	7.06
5	Heat flow density	7.16	44.10	3.98
6	Heating system	0.92	1.35	0.36

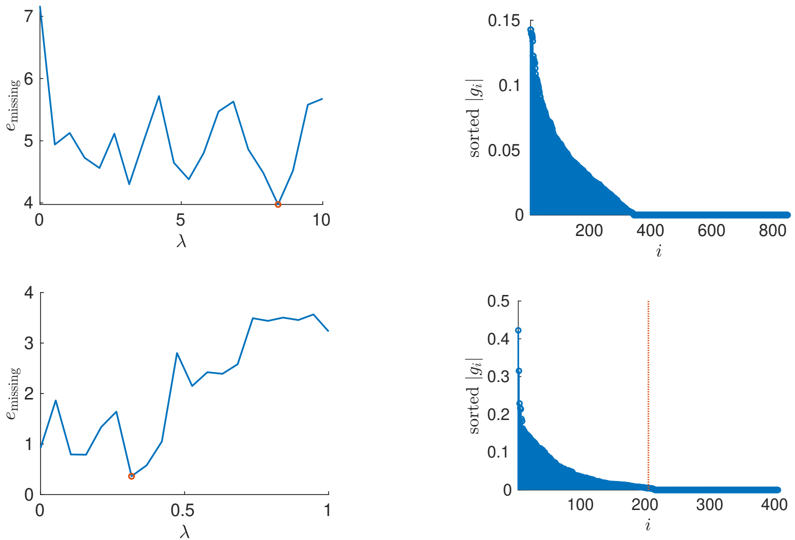
Tuning of λ and sparsity of g (datasets 1, 2)



Tuning of λ and sparsity of g (datasets 3, 4)



Tuning of λ and sparsity of g (datasets 5, 6)



Summary: convex relaxations

w_d exact $\rightsquigarrow$ system theory

- exact analytical solution
- current work: efficient real-time algorithms

w_d inexact $\rightsquigarrow$ nonconvex optimization

- subspace methods
- local optimization
- convex relaxations

empirical validation

- the naive approach works (surprisingly) well
- parametric local optimization is not robust
- ℓ_1-norm regularization gives the best results

Meta conclusions

critical attitude

- ask questions (and search for answers)
- don't trust authorities, instead rediscover
- new ideas start with bothersome inconsistencies

theory–algorithms synergy

- useful ideas lead to algorithms
- algorithms clarify and refine the ideas
- software makes the theory practically useful

rigor vs intuition

- hard real-life problems rarely admit rigorous solutions
- watch out for hidden / unverifiable assumptions
- the ℓ_1-norm heuristic is unreasonably effective

Outline

Pedagogical example: Free fall prediction

Case study: Dynamic measurement

Outline

Pedagogical example: Free fall prediction

Case study: Dynamic measurement

The goal is to predict free fall trajectory without knowing the laws of physics

object with mass m, falling in gravitational field

- y — position
- $v := \dot{y}$ — velocity
- $y(0), v(0)$ — initial condition

task: given initial condition, find the trajectory y

- model-based approach:
  1. physics $\mapsto$ model
  2. model + ini. cond. $\mapsto y$
- data-driven approach: data $y_\mathrm{d}^1, \dots, y_\mathrm{d}^N$ + ini. cond. $\mapsto y$

Modeling from first principles leads to affine time-invariant state-space model

second law of Newton + the law of gravity

$m\ddot{y} = m \begin{bmatrix} 0 \\ 9.81 \end{bmatrix} + f$, where $y(0) = y_{\text{ini}}$ and $\dot{y}(0) = v_{\text{ini}}$

- 9.81 — gravitational constant
- $f = -\gamma v$ — force due to friction in the air

state $x := (y_1, \dot{y}_1, y_2, \dot{y}_2, x_5)$, where $x_5 = -9.81$

initial state $x_{\text{ini}} := (y_{\text{ini},1}, v_{\text{ini},1}, y_{\text{ini},2}, v_{\text{ini},2}, -9.81)$

Modeling from first principles leads to affine time-invariant state-space model

$$\dot{x} = \begin{bmatrix} 0 & 1 & & & \\ 0 & -\gamma/m & & & \\ & & 0 & 1 & \\ & & 0 & -\gamma/m & 1 \\ & & & & 0 \end{bmatrix} x, \qquad x(0) = \begin{bmatrix} y_{\text{ini},1} \\ v_{\text{ini},1} \\ y_{\text{ini},2} \\ v_{\text{ini},2} \\ -9.81 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} x$$

data: N, T-samples long discretized trajectories

Simulation setup and data

write a function `fall` that simulates free fall

```
y = fall(y0, v0, t, m, gamma)
```

simulate `N=10`, `T=100`-samples long trajectories

```
m = 1; gamma = 0.5;
N = 10; T = 100; t = linspace(0, 1, T);
for i = 1:N,
  y{i} = fall(rand(2,1), rand(2,1), t,gamma,m);
end
```

and to-be-predicted trajectory

```
y_new = fall(rand(2,1), rand(2,1), t,gamma,m);
```

Data-driven free fall prediction method

data "informativity" condition:

$$\text{rank} \underbrace{\begin{bmatrix} y_\text{d}^1 & \cdots & y_\text{d}^N \end{bmatrix}}_{D} = 5$$

algorithm for data-driven prediction:

1. solve $\begin{bmatrix} y_\text{d}^1(1) & \cdots & y_\text{d}^N(1) \\ y_\text{d}^1(2) & \cdots & y_\text{d}^N(2) \\ y_\text{d}^1(3) & \cdots & y_\text{d}^N(3) \end{bmatrix} g = \underbrace{\begin{bmatrix} y(1) \\ y(2) \\ y(3) \end{bmatrix}}_{\text{ini. cond.}}$

2. define $y := Dg$

Verify that the data-driven prediction "works"

check the data "informativity" condition

```
[rank(D) rank([vec(y_new') D])] % -> [ 5 5 ]
```

implement the data-driven computation method

verify the computed solution

Summary: prediction of free fall trajectory

first principles modeling

- use the second law of Newton and the law of gravity
- in particular, the Earth's gravitational constant is used
- lead to an autonomous affine time-invariant system

data-driven methods

- bypass the knowledge of the physical laws
- automatically infer and use them
- no hyper-parameters to tune

Outline

Pedagogical example: Free fall prediction

Case study: Dynamic measurement

My interest in dynamic measurement started from a textbook problem

"A thermometer reading $21^\circ C$*, which has been inside a house for a long time, is taken outside. After one minute the thermometer reads* $15^\circ C$*; after two minutes it reads* $11^\circ C$*. What is the outside temperature?"*

According to Newton's law of cooling, an object of higher temperature than its environment cools at a rate that is proportional to the difference in temperature.

Main idea: predict the steady-state value from the first few samples of the transient

textbook problem:

- 1st order dynamics
- 3 noise-free samples
- batch solution

generalizations:

- $n \geq 1$ order dynamics
- $T \geq 3$ noisy (vector) samples
- recursive computation

implementation and practical validation

Thermometer: first order dynamical system

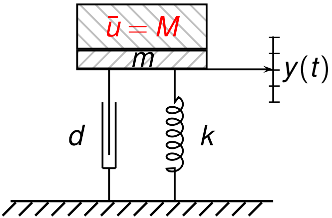
$$\begin{array}{c} \text{environmental} \\ \text{temperature } \bar{u} \end{array} \xrightarrow{\text{heat transfer}} \begin{array}{c} \text{thermometer's} \\ \text{reading } y \end{array}$$

measurement process: Newton's law of cooling

$$y = a\left(\bar{u} - y\right)$$

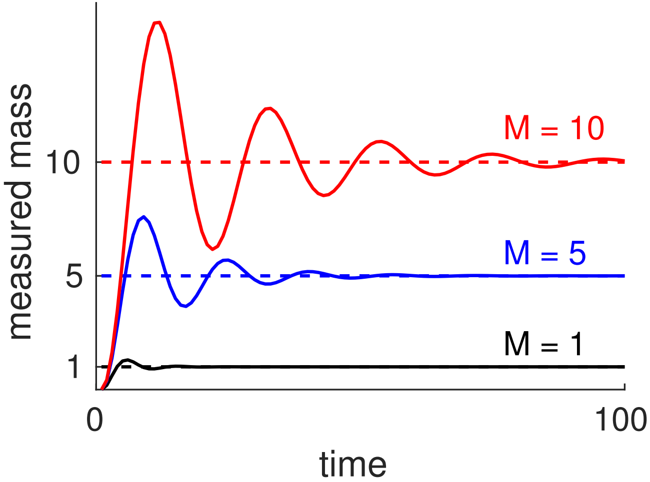
heat transfer coefficient $a > 0$

Scale: second order dynamical system



$$(M+m)\frac{\mathrm{d}}{\mathrm{d}\,t}y + dy + ky = g\bar{u}$$

The measurement process dynamics depends on the to-be-measured mass



Dynamic measurement: take into account the dynamical properties of the sensor

$$\begin{array}{c}\text{to-be-measured}\\ \text{variable } u\end{array} \xrightarrow{\text{measurement process}} \begin{array}{c}\text{measured}\\ \text{variable } y\end{array}$$

assumption 1: measured variable is constant $u(t) = \bar{u}$

assumption 2: the sensor is stable LTI system

assumption 3: sensor's DC-gain $= 1$ (calibrated sensor)

The data is generated from LTI system with output noise and constant input

$$\underbrace{y_\mathrm{d}}_{\substack{\text{measured}\\ \text{data}}} = \underbrace{y}_{\substack{\text{true}\\ \text{value}}} + \underbrace{e}_{\substack{\text{measurement}\\ \text{noise}}}$$

$$\underbrace{y}_{\substack{\text{true}\\ \text{value}}} = \underbrace{\bar{u}}_{\substack{\text{steady-state}\\ \text{value}}} + \underbrace{y_0}_{\substack{\text{transient}\\ \text{response}}}$$

assumption 4: e is a zero mean, white, Gaussian noise using a state space representation of the sensor

$$x(t+1) = Ax(t), \qquad x(0) = x_0$$
$$y_0(t) = cx(t)$$

we obtain

$$\underbrace{\begin{bmatrix} y_{\mathrm{d}}(1) \\ y_{\mathrm{d}}(2) \\ \vdots \\ y_{\mathrm{d}}(T) \end{bmatrix}}_{y_{\mathrm{d}}} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}}_{\mathbf{1}_T} \bar{u} + \underbrace{\begin{bmatrix} c \\ cA \\ \vdots \\ cA^{T-1} \end{bmatrix}}_{\mathscr{O}_T} x_0 + \underbrace{\begin{bmatrix} e(1) \\ e(2) \\ \vdots \\ e(T) \end{bmatrix}}_{e}$$

Maximum-likelihood model-based estimator

solve approximately

$$\begin{bmatrix} \mathbf{1}_T & \mathscr{O}_T \end{bmatrix} \begin{bmatrix} \widehat{u} \\ \widehat{x}_0 \end{bmatrix} \approx y_\mathrm{d}$$

standard least-squares problem

$$\begin{array}{ll} \text{minimize} & \text{over } \widehat{y}, \widehat{u}, \widehat{x}_0 \quad \|y_\mathrm{d} - \widehat{y}\| \\ \text{subject to} & \begin{bmatrix} \mathbf{1}_T & \mathscr{O}_T \end{bmatrix} \begin{bmatrix} \widehat{u} \\ \widehat{x}_0 \end{bmatrix} = \widehat{y} \end{array}$$

recursive implementation $\rightsquigarrow$ Kalman filter

Subspace model-free method

goal: avoid using the model parameters (A, C, $\mathscr{O}_T$)

in the noise-free case, due to the LTI assumption,

$$\Delta y(t) := y(t) - y(t-1) = y_0(t) - y_0(t-1)$$

satisfies the same dynamics as y_0, *i.e.*,

$$\begin{aligned} x(t+1) &= Ax(t), \qquad x(0) = \Delta x \\ \Delta y(t) &= cx(t) \end{aligned}$$

Hankel matrix—construction of multiple "short" trajectories from one "long" trajectory

$$\mathscr{H}(\Delta y) := \begin{bmatrix} \Delta y(1) & \Delta y(2) & \cdots & \Delta y(\mathtt{n}) \\ \Delta y(2) & \Delta y(3) & \cdots & \Delta y(\mathtt{n}+1) \\ \Delta y(3) & \Delta y(4) & \cdots & \Delta y(\mathtt{n}+2) \\ \vdots & \vdots & & \vdots \\ \Delta y(T-\mathtt{n}) & \Delta y(T-\mathtt{n}) & \cdots & \Delta y(T-1) \end{bmatrix}$$

fact: if $\operatorname{rank} \mathscr{H}(\Delta y) = \mathtt{n}$, then

$$\operatorname{image} \mathscr{O}_{T-\mathtt{n}} = \operatorname{image} \mathscr{H}(\Delta y)$$

model-based equation

$$\begin{bmatrix} \mathbf{1}_T & \mathscr{O}_T \end{bmatrix} \begin{bmatrix} \bar{u} \\ \widehat{x}_0 \end{bmatrix} = y$$

data-driven equation

$$\begin{bmatrix} \mathbf{1}_{T-\mathtt{n}} & \mathscr{H}(\Delta y) \end{bmatrix} \begin{bmatrix} \bar{u} \\ \ell \end{bmatrix} = y|_{T-\mathtt{n}} \qquad (*)$$

subspace method

solve $(*)$ by (recursive) least squares

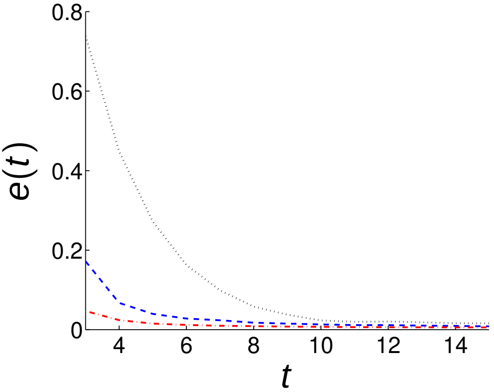
Empirical validation

dashed	—	true parameter value $\bar{u}$
solid	—	true output trajectory y_0
dotted	—	naive estimate $\widehat{u} = G^{+}y$
dashed	—	model-based Kalman filter
bashed-dotted	—	data-driven method

estimation error: $e := \frac{1}{N}\sum_{i=1}^{N} \|\bar{u} - \widehat{u}^{(i)}\|$

(for $N = 100$ Monte-Carlo repetitions)

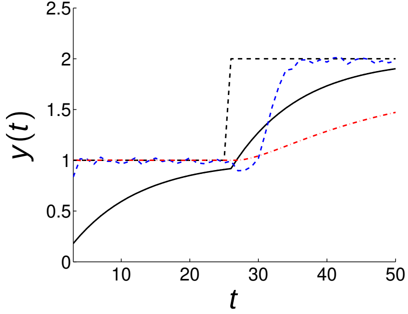
Simulated data of dynamic cooling process

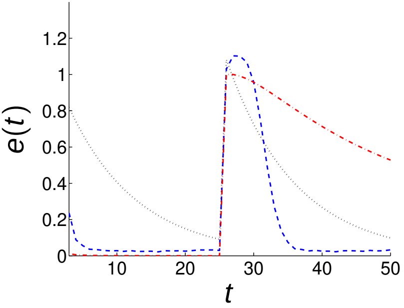


$e(t) \to 0$ as $t \to \infty$ at different rates

best is the Kalman filter (maximum likelihood estimator)

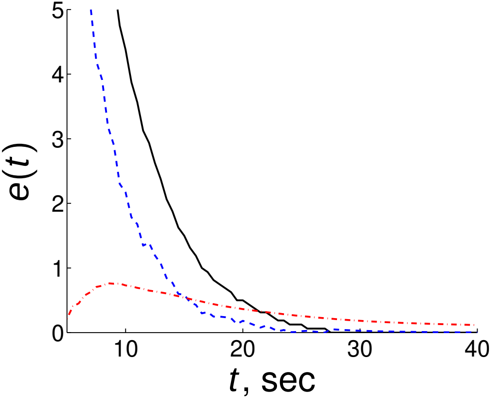
Simulation with time-varying parameter





Proof of concept prototype

Results in real-life experiment



Summary

dynamic measurement

steady-state value prediction

the subspace method is applicable for

- high order dynamics
- noisy vector observations
- online computation

future work / open problems

- numerical efficiency
- real-time uncertainty quantification
- generalization to nonlinear systems