Combining Linear Algebra and Numerical Optimization for Gray-Box Affine State-Space Model Identification

Guillaume Mercère and Olivier Prot

University of Poitiers

MIRES meeting
13 April 2022

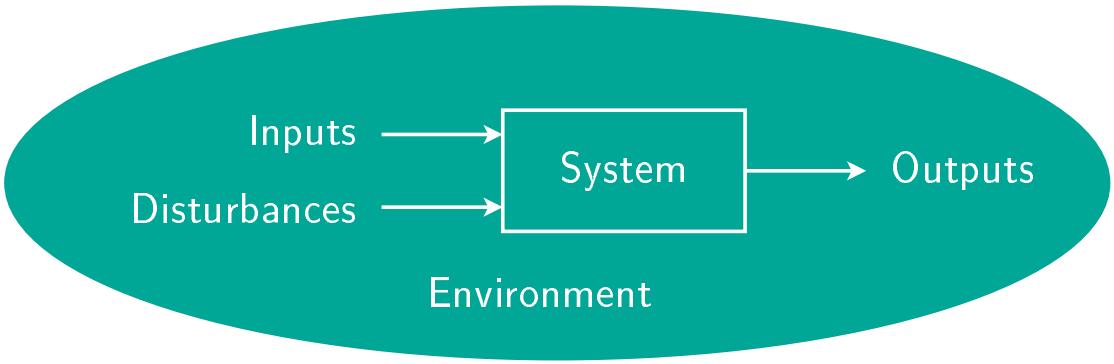
Outline

1. Data driven model learning
2. Gray box model learning
   - A standard solution
   - A linear algebra based solution
3. Conclusions and discussion

DATA DRIVEN MODEL LEARNING

System and model

- A system can be seen as an organized group of interacting components the engineer wants to describe and/or understand the behavior of.
- A system is surrounded by its environment.
- The interactions between the system and its environment pass through its inputs, its outputs and disturbances.



- Models are mainly mathematical (and simplified?) representations used for describing and understanding these interactions accurately.

Modeling

- Modeling physical systems accurately is a paramount step within most of the engineering sciences.
- Models can be viewed as a smart embedding (into a compact mathematical description) of engineer's knowledge, physical facts, intuitive insights as well as information available in measured signals.
- Models can be used for simulation, prediction, diagnosis, controller design,...
- The complexity of a model is linked to
  - the complexity of the system to be modeled,
  - the future use of this model.

"All models are wrong[1]*, but some are useful."*

[1] Box, G. E. P. (1976), "Science and statistics", Journal of the American Statistical Association, 71 , pp. 791-799

Data driven modeling

- We focus on methods for modeling the dynamical behavior of systems involving engineer's prior knowledge and data mainly.
- Data driven model learning requires
  - a model class, *i.e.*, a set of model candidates often parameterized by an unknown (finite) parameter vector,
  - data sets containing enough information to select "a good" model among the candidates in the model class,
  - a (scalar) fitting measurement to be optimized in order to yield the parameter vector which gives the best mimicking capacities of the estimated model,
  - a model validation step which guarantees that the estimated model satisfies the constraints and conditions it was built for.

Université de Poitiers

State space model

- As far as the model class is concerned, we focus on state space models, *i.e.*,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}),$$
$$\boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}),$$

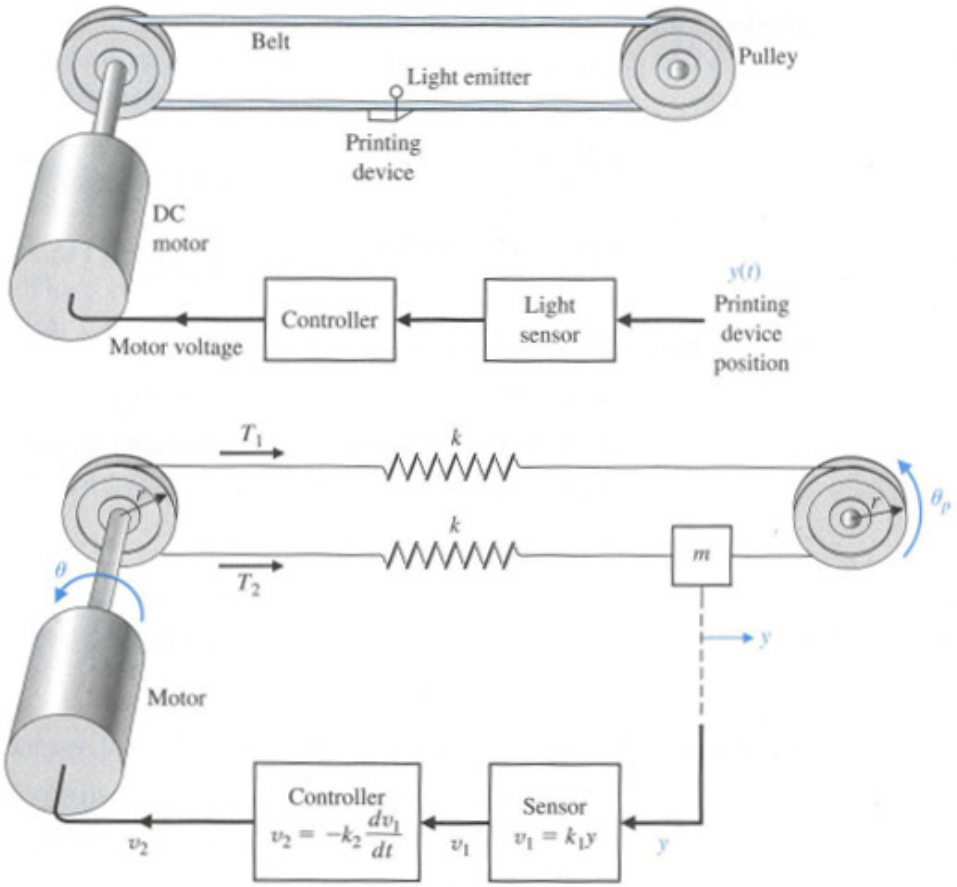
  where
  - $\boldsymbol{x}(t) \in \mathbb{R}^{n_x \times 1}$ is the state vector,
  - $\boldsymbol{y}(t) \in \mathbb{R}^{n_y \times 1}$ is the output vector,
  - $\boldsymbol{u}(t) \in \mathbb{R}^{n_u \times 1}$ is the input vector (controlled by the user),
  - $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta \times 1}$ is the vector of unknown parameters to be estimated from the available data sets,
  - $\boldsymbol{f} : \mathbb{R}^{n_x \times 1} \times \mathbb{R}^{n_u \times 1} \times \mathbb{R}^{n_\theta \times 1} \to \mathbb{R}^{n_x \times 1}$ is a user defined smooth multivariate function,
  - $\boldsymbol{g} : \mathbb{R}^{n_x \times 1} \times \mathbb{R}^{n_u \times 1} \times \mathbb{R}^{n_\theta \times 1} \to \mathbb{R}^{n_y \times 1}$ is a user defined smooth multivariate function.

Université de Poitiers

Fifty shades of gray

- According to the required complexity level, different shades of gray are invoked to describe the models and the prior starting from white to black.
- White box models are the results of an extensive physical modeling step from first principles and prior knowledge.
- Model with lightest shades of gray can be obtained when some parameters of the white box models are unknown or uncertain.
- Black box models are introduced for system behavior description with user defined flexible functions (to be fitted to the available data) when no physical prior is available.

Toy example



Picture from "Modern control systems", 2010, pp. 222–228.

Toy example (cont'd)

- By assuming that this system is linear and time invariant (LTI), an easy way to describe its dynamical behavior consists in resorting to a black box LTI model

$$\dot{\mathfrak{x}}(t) = \mathfrak{A}\mathfrak{x}(t) + \mathfrak{B}u(t),$$
$$y(t) = \mathfrak{C}\mathfrak{x}(t),$$

where

  - $y(t)$ the speed of the printing device,
  - $u(t)$ the torque applied to drive the belt,

whereas $(\mathfrak{A}, \mathfrak{B}, \mathfrak{C})$ are fully parameterized and without any physical meaning (like $\mathfrak{x}(t)$).

Toy example (cont'd)

- By using Newton's laws, we can prove that this system satisfies

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x}(t) + \boldsymbol{B}(\boldsymbol{\theta})u(t),$$
$$y(t) = \boldsymbol{C}(\boldsymbol{\theta})\boldsymbol{x}(t),$$

with $\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) & x_2(t) & x_3(t) \end{bmatrix}^\top$ and

$$x_1(t) = r\phi(t) - z(t), \quad x_2(t) = \dot{z}(t), \quad x_3(t) = \dot{\phi}(t),$$

and

  - $y(t)$ the speed of the printing device,
  - $u(t)$ the torque applied to drive the belt.

Toy example (cont'd)

- By using Newton's laws, we can prove that this system satisfies

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x}(t) + \boldsymbol{B}(\boldsymbol{\theta})u(t),$$
$$y(t) = \boldsymbol{C}(\boldsymbol{\theta})\boldsymbol{x}(t),$$

with

$$\boldsymbol{A}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & -1 & \theta_1 \\ \theta_2 & 0 & 0 \\ \theta_3 & 0 & \theta_4 \end{bmatrix}, \qquad \boldsymbol{B}(\boldsymbol{\theta}) = \begin{bmatrix} 0 \\ 0 \\ \theta_5 \end{bmatrix},$$
$$\boldsymbol{C}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}.$$

Toy example (cont'd)

- By using Newton's laws, we can prove that this system satisfies

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x}(t) + \boldsymbol{B}(\boldsymbol{\theta})u(t),$$
$$y(t) = \boldsymbol{C}(\boldsymbol{\theta})\boldsymbol{x}(t),$$

with

$$\theta_1 = r, \qquad \theta_2 = \frac{2k}{m}, \qquad \theta_3 = -\frac{2kr}{J},$$
$$\theta_4 = -\frac{b}{J}, \qquad \theta_5 = -\frac{k_m}{RJ}.$$

GRAY BOX MODEL LEARNING

A STANDARD SOLUTION

Output error method

- In order to estimate the parameter vector $\boldsymbol{\theta}$ from available data $\{u(k), y_m(k)\}_{k=0}^{N-1}$, a standard solution consists in
  - fixing a model structure

$$\boldsymbol{A}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & -1 & \theta_1 \\ \theta_2 & 0 & 0 \\ \theta_3 & 0 & \theta_4 \end{bmatrix}, \qquad \boldsymbol{B}(\boldsymbol{\theta}) = \begin{bmatrix} 0 \\ 0 \\ \theta_5 \end{bmatrix},$$

$$\boldsymbol{C}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix},$$

Output error method

- In order to estimate the parameter vector $\boldsymbol{\theta}$ from available data $\{u(k), y_m(k)\}_{k=0}^{N-1}$, a standard solution consists in
  - fixing a model structure
  - checking the identifiability

    We must assume, *e.g.*, that θ_1 is known to guarantee that the model structure

$$\boldsymbol{A}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & -1 & 0.15 \\ \theta_2 & 0 & 0 \\ \theta_3 & 0 & \theta_4 \end{bmatrix}, \qquad \boldsymbol{B}(\boldsymbol{\theta}) = \begin{bmatrix} 0 \\ 0 \\ \theta_5 \end{bmatrix},$$

$$\boldsymbol{C}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix},$$

    becomes identifiable.

Output error method

- In order to estimate the parameter vector $\boldsymbol{\theta}$ from available data $\{u(k), y_m(k)\}_{k=0}^{N-1}$, a standard solution consists in
  - fixing a model structure
  - checking the identifiability
  - comparing with a signal norm the outputs of the system and the model, *e.g.*,

$$V_N(\boldsymbol{\theta}) = \frac{1}{N} \sum_{k=0}^{N-1} \|y_m(k) - y(k, \boldsymbol{\theta})\|_{\boldsymbol{Q}}^2 .$$

  - minimizing $V_N(\boldsymbol{\theta})$ can be efficiently performed by using, *e.g.*, the Levenberg-Marquardt or BFGS algorithms.

Simulation example

- Simulation procedure:
  - one noise-free data-set for the identification (PRBS)
  - another noise-free data-set for the validation (PRBS)
  - A double Monte Carlo simulation is run

Simulation example

- Simulation procedure:
  - one noise-free data-set for the identification (PRBS)
  - another noise-free data-set for the validation (PRBS)
  - A double Monte Carlo simulation is run
    - for the noise, 100 (resp. 100) zero-mean white (resp. colored) Gaussian noises are generated such that the SNR $= 10\ dB$,

Simulation example

- Simulation procedure:
  - one noise-free data-set for the identification (PRBS)
  - another noise-free data-set for the validation (PRBS)
  - A double Monte Carlo simulation is run
    - for the noise, 100 (resp. 100) zero-mean white (resp. colored) Gaussian noises are generated such that the SNR $= 10\ dB$,
    - for the initialization,

$$\theta_i^{init} = \theta_i^{real}(0.5 - \lambda),$$

where λ is a random value drawn from the standard uniform distribution on the open interval $(0, 1)$.

Param. estimation results

Zero-mean white Gaussian noise

SNR	10dB			
Param	θ_2 (200)	θ_3 (-600)	θ_4 (-25)	θ_5 (100)
Max	1.98e+03	-517.33	-22.44	1.62e+03
Min	171.78	-2.5e+03	-271.45	28.58
25 percentile	195.47	-622.52	-25.67	97.02
75 percentile	205.25	-575.94	-24.39	102.99
Median	201.11	-597.39	-25.06	99.21
Mean	202.51	-618.07	-25.37	98.l4

Param. estimation results

Zero-mean colored Gaussian noise

SNR	10dB			
Param	θ_2 (200)	θ_3 (-600)	θ_4 (-25)	θ_5 (100)
Max	4.205e+03	-229.64	-17.85	3.38e+03
Min	143.70	-3.191e+03	-535.00	30.09
25 percentile	185.94	-659.11	-27.86	97.06
75 percentile	219.48	-526.46	-23.22	103.1
Median	199.83	-598.68	-24.86	101.89
Mean	209.16	-583.29	-26.11	102.488

Université de Poitiers

I/O and frequency fits

Reminder

$$\mathsf{BFT} = 100 \times \max\left(1 - \frac{\|y - \gamma\|_2^2}{\|y - \mathsf{mean}(y)\|_2^2}, 0\right),$$

$$H_\infty = \|H(\boldsymbol{\theta}, s) - H(\hat{\boldsymbol{\theta}}, s)\|_\infty$$

I/O and frequency fits

		White noise	Colored noise
	SNR	$10dB$	$10dB$
BFT	Best	99.66	99.75
	Worst	74.61	74.55
	25 percentile	98.57	94.14
	75 percentile	99.17	96.78
	Median	99.92	95.67
	Mean	98.92	95.27
H_∞	Best	0.0207	0.2708
	Worst	2.9554	2.9554
	25 percentile	0.0594	0.3795
	75 percentile	0.1653	1.0383
	Median	0.0974	0.5362
	Mean	0.7342	1.0602

Complementary analysis

- Mean computational time: $0.73\ s$.
- Convergence success rate: $76.2\ \%$ (white) and $73.2\ \%$ (colored).

Complementary analysis

- Mean computational time: $0.73\ s$.
- Convergence success rate: $76.2\ \%$ (white) and $73.2\ \%$ (colored).
- To sum up:
  - non convex cost function
    - initialization problem,
    - "time consuming",
  - iterative algorithm requiring simulation of models
    - unstable "local" models can pop up.
  - Dealing with non standard disturbances (EIV, ...) can be tricky.
- Question: can you outperform the output error method?

Université de Poitiers

I/O and frequency fits: black-box models

SRIVC

		White noise	Colored noise
	SNR	$10dB$	$10dB$
BFT	Best	99.73	98.97
	Worst	97.91	90.16
	25 percentile	99.69	94.48
	75 percentile	99.23	96.79
	Median	98.95	95.81
	Mean	98.94	95.63
H_∞	Best	0.0208	0.0838
	Worst	0.1831	1.0483
	25 percentile	0.0504	0.2903
	75 percentile	0.0999	0.5224
	Median	0.0698	0.3988
	Mean	0.0788	0.4176

A LINEAR ALGEBRA BASED SOLUTION

Problem formulation

- Let use assume that
  - We have sufficiently rich and long data sets.
  - A reliable fully parameterized state space model $(\mathfrak{A}, \mathfrak{B}, \mathfrak{C})$ is available.
  - The parameterization $(\boldsymbol{A}(\boldsymbol{\theta}), \boldsymbol{B}(\boldsymbol{\theta}), \boldsymbol{C}(\boldsymbol{\theta}))$ is identifiable.
- We know that matrices $\boldsymbol{T}$ or $\boldsymbol{S}$ exist such that

$$\mathfrak{A}\boldsymbol{T} = \boldsymbol{T}\boldsymbol{A}(\boldsymbol{\theta}), \quad \mathfrak{B} = \boldsymbol{T}\boldsymbol{B}(\boldsymbol{\theta}), \quad \mathfrak{C}\boldsymbol{T} = \boldsymbol{C}(\boldsymbol{\theta}),$$
$$\boldsymbol{S}\mathfrak{A} = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{S}, \quad \boldsymbol{S}\mathfrak{B} = \boldsymbol{B}(\boldsymbol{\theta}), \quad \mathfrak{C} = \boldsymbol{C}(\boldsymbol{\theta})\boldsymbol{S}.$$

Problem to solve

Determine the matrices $\boldsymbol{T}$ or $\boldsymbol{S}$ (and by extension $\boldsymbol{\theta}$) by solving the aforementioned set of equations.

Another toy example

- Let us consider the gray box model ($\theta_1 = -1$ and $\theta_2 = -2$ when necessary)

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 1 & \theta_1 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} \theta_2 \\ 3 \end{bmatrix} \boldsymbol{u}(t),$$
$$\boldsymbol{y}(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}(t).$$

- Let us assume that

$$\mathfrak{A} = \begin{bmatrix} 2.1961e{+}01 & -5.9101e{+}01 \\ 7.4341 & -1.9961e{+}01 \end{bmatrix}, \quad \mathfrak{B} = \begin{bmatrix} 3.6753 \\ 1.3522 \end{bmatrix},$$
$$\boldsymbol{T} = \begin{bmatrix} 9.5974e{-}01 & 5.8527e{-}01 \\ 3.4039e{-}01 & 2.2381e{-}01 \end{bmatrix}, \quad \mathfrak{C}^\top = \begin{bmatrix} 1.4360e{+}01 \\ -3.7552e{+}01 \end{bmatrix}.$$

Another toy example (cont'd)

- Instead of focusing on the unknowns θ_1 and θ_2, we concentrate on the known parameters, *i.e.*,

$$\boldsymbol{S}\mathfrak{A} = \begin{bmatrix} 1 & \theta_1 \\ 0 & 1 \end{bmatrix} \boldsymbol{S}, \qquad \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{S}\mathfrak{A} = \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{S},$$

$$\boldsymbol{S}\mathfrak{B} = \begin{bmatrix} \theta_2 \\ 3 \end{bmatrix}, \qquad \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{S}\mathfrak{B} = 3$$

$$\mathfrak{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{S} \qquad \mathfrak{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{S}.$$

- Written differently, we get

$$\begin{bmatrix} \mathfrak{A}^\top \otimes \boldsymbol{I}_{2\times 2} - \boldsymbol{I}_{2\times 2} \otimes \begin{bmatrix} 0 & 1 \end{bmatrix} \\ \boldsymbol{I}_{2\times 2} \otimes \begin{bmatrix} 1 & 0 \end{bmatrix} \\ \mathfrak{B}^\top \otimes \boldsymbol{I}_{2\times 2} \end{bmatrix} \mathrm{vec}(\boldsymbol{S}) = \begin{bmatrix} \boldsymbol{0}_{1\times 2} \\ \mathrm{vec}(\mathfrak{C}) \\ 3 \end{bmatrix}.$$

Another toy example (cont'd)

- $\boldsymbol{S}$ is obtained by solving this system of linear equations, *i.e.*,

$$\boldsymbol{S} = \begin{bmatrix} 1.4360e{+}01 & -3.7552e{+}01 \\ -2.1840e{+}01 & 6.1580e{+}01 \end{bmatrix} = \boldsymbol{T}^{-1}.$$

- Our idea is to extend this approach when
  - the dependence on $\boldsymbol{\theta}$ is affine,
  - the system to identify is in the model class.

Another toy example (cont'd)

- $\boldsymbol{S}$ is obtained by solving this system of linear equations, *i.e.*,

$$\boldsymbol{S} = \begin{bmatrix} 1.4360e{+}01 & -3.7552e{+}01 \\ -2.1840e{+}01 & 6.1580e{+}01 \end{bmatrix} = \boldsymbol{T}^{-1}.$$

- Our idea is to extend this approach when
  - the dependence on $\boldsymbol{\theta}$ is affine,
  - the system to identify is in the model class.
- More specifically, we have developed a solution which
  - extracts a large system of equations linear in terms of the similarity transformation,
  - solves this system of linear equations by using standard linear algebra tools,
  - completes this two step procedure by a numerical optimization solution (dedicated now to a very small number of unknowns) if necessary.

Université de Poitiers

Explanations with the printer belt

- We consider the model parameterization

$$\boldsymbol{A}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & -1 & \theta_1 \\ \theta_2 & 0 & 0 \\ \theta_3 & 0 & \theta_4 \end{bmatrix}, \qquad \boldsymbol{B}(\boldsymbol{\theta}) = \begin{bmatrix} 0 \\ 0 \\ \theta_5 \end{bmatrix},$$
$$\boldsymbol{C}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}.$$

- We know (in red)

$$\boldsymbol{A}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & -1 & \theta_1 \\ \theta_2 & 0 & 0 \\ \theta_3 & 0 & \theta_4 \end{bmatrix}, \qquad \boldsymbol{B}(\boldsymbol{\theta}) = \begin{bmatrix} 0 \\ 0 \\ \theta_5 \end{bmatrix},$$
$$\boldsymbol{C}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}.$$

Université de Poitiers

Explanations with the printer belt (cont'd)

- Thanks to the insight on four rows in $(\boldsymbol{A}(\boldsymbol{\theta}), \boldsymbol{B}(\boldsymbol{\theta}), \boldsymbol{C}(\boldsymbol{\theta}))$ as well as the availability of $(\mathfrak{A}, \mathfrak{B}, \mathfrak{C})$, we try to estimate $\boldsymbol{S}$ via the set of linear equations

$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \boldsymbol{S}\mathfrak{A} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{S},$$
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \boldsymbol{S}\mathfrak{B} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$
$$\mathfrak{C} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \boldsymbol{S}.$$

Explanations with the printer belt (cont'd)

- By using the vectorization tool

$$\underbrace{\begin{bmatrix} \mathfrak{A}^\top \otimes \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} - \boldsymbol{I}_{3\times 3} \otimes \begin{bmatrix} 0 & -1 & \theta_1 \end{bmatrix} \\ \mathfrak{B}^\top \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\ \boldsymbol{I}_{3\times 3} \otimes \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \end{bmatrix}}_{\boldsymbol{N}} \mathrm{vec}(\boldsymbol{S}) = \underbrace{\begin{bmatrix} \mathbf{0}_{3\times 1} \\ \mathbf{0}_{2\times 1} \\ \mathrm{vec}(\mathfrak{C}) \end{bmatrix}}_{\boldsymbol{\alpha}}$$

- With $\boldsymbol{N} \in \mathbb{R}^{8\times 9}$, we have 8 equations for 9 unknown parameters in $\boldsymbol{S}$.
- Because $\ker(\boldsymbol{N}) \neq \{\mathbf{0}\}$, the solution of this set of equations is not unique!
- We can however compute a (non unique) solution of it.

Explanations with the printer belt (cont'd)

- We can compute $\ker(\boldsymbol{N})$ and show that $\ker(\boldsymbol{N}) \in \mathbb{R}^{9\times 2}$.
- Furthermore,

$$\text{vec}\left(\hat{\boldsymbol{S}}\right) = \boldsymbol{N}^{\dagger}\left(\boldsymbol{\alpha} + \boldsymbol{\alpha}_{\text{null}}\right),$$

  where $\boldsymbol{\alpha}_{\text{null}}$ stands for any vector belonging to $\ker(\boldsymbol{N})$.
- In order to bypass the ambiguities relevant to $\ker(\boldsymbol{N})$, we can use the prior on $\boldsymbol{A}(\boldsymbol{\theta})$ unused until now.

Université de Poitiers

Explanations with the printer belt (cont'd)

- We introduce the cost function

$$\arg\min_{\boldsymbol{z}\in\mathbb{R}^{2\times 1}} \left\| \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{\mathcal{A}}(\boldsymbol{z}) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\|_2^2$$

$$\text{s.t.} \begin{cases} \boldsymbol{\mathcal{A}}(\boldsymbol{z}) = \boldsymbol{\mathcal{S}}(\boldsymbol{z})\mathfrak{A}\boldsymbol{\mathcal{S}}^{-1}(\boldsymbol{z}) \\ \boldsymbol{\mathcal{S}}(\boldsymbol{z}) = \text{reshape}\left(\boldsymbol{N}^{\dagger}\boldsymbol{\alpha} + \ker\left(\boldsymbol{N}\right)\boldsymbol{z}, n_x, n_x\right) \end{cases}.$$

- Thanks to linear algebra, we have to determine 2 unknown parameters instead of 4.
- This cost function can be minimized with a L-BFGS algorithm without strong prior.

Parameter estimation results

- We use (again) the 100 noisy data sets to estimate 100 black box models by using SRIVC.
- We select the best black box model giving the best fit on a validation data set.
- The matrices $(\mathfrak{A}, \mathfrak{B}, \mathfrak{C})$ are built by balancing the best SRIVC model.
- The initialization of the BFGS algorithm is performed with 100 different initial vectors z randomly generated from a standard zero-mean normal distribution.
- The best z is chosen by selecting the $\hat{S}$ having the best condition number.

Parameter estimation results (cont'd)

- We can compute $(\boldsymbol{A}(\boldsymbol{\theta}), \boldsymbol{B}(\boldsymbol{\theta}), \boldsymbol{C}(\boldsymbol{\theta}))$ from $(\mathfrak{A}, \mathfrak{B}, \mathfrak{C})$ and $\hat{\boldsymbol{S}}$.

	NOMINAL	ESTIMATED
θ_2	200	201.4
θ_3	-600	-602.2
θ_4	-25	-23.5
θ_5	100	99.2

Explanations with the printer belt (cont'd)

- Let us assume that θ_2 is known as well.
- Because θ_2 is now known *a priori*, we know the first two rows of $\boldsymbol{A}(\boldsymbol{\theta})$.
- By combining this prior with the knowledge of $\boldsymbol{C}(\boldsymbol{\theta})$, we get

$$\underbrace{\begin{bmatrix} \mathfrak{A}^\top \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} - \boldsymbol{I}_{3\times 3} \otimes \begin{bmatrix} 0 & -1 & \theta_1 \\ \theta_2 & 0 & 0 \end{bmatrix} \\ \boldsymbol{I}_{3\times 3} \otimes \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \end{bmatrix}}_{\boldsymbol{N}} \times \operatorname{vec}(\boldsymbol{S}) = \underbrace{\begin{bmatrix} \mathbf{0}_{6\times 1} \\ \operatorname{vec}(\mathfrak{C}) \end{bmatrix}}_{\boldsymbol{\alpha}}.$$

- Now $\boldsymbol{N} \in \mathbb{R}^{9\times 9}$ and $\ker(\boldsymbol{N}) = \{\mathbf{0}\}$!

Parameter estimation results (cont'd)

- We can compute $\hat{\boldsymbol{S}}$ without any nonlinear optimization!!!
- Again, $(\boldsymbol{A}(\boldsymbol{\theta}), \boldsymbol{B}(\boldsymbol{\theta}), \boldsymbol{C}(\boldsymbol{\theta}))$ from $(\mathfrak{A}, \mathfrak{B}, \mathfrak{C})$ and $\hat{\boldsymbol{S}}$.

	NOMINAL	ESTIMATED
θ_3	-600	-601.3
θ_4	-25	-24.4
θ_5	100	99.1

CONCLUSIONS AND DISCUSSION

Take home message

- The problem of parameterized LTI model learning has been studied.
- A solution combining linear algebra and numerical optimization has been introduced.
- Experiments show that using this new solution to get reliable initial guesses for an output error method is efficient.
- Extension to solutions combining equations on $\boldsymbol{S}$ and $\boldsymbol{T}$ is under progress (keep in mind that $\boldsymbol{S} = \boldsymbol{T}^{-1}$).

Université de Poitiers