Revisiting gray box model learning and Kalman filtering with subspace based model learning

Guillaume Mercère

University of Poitiers

KU Leuven, 11 June 2024

Outline

1. Motivations
2. Subspace based model learning (SML): a reminder
3. Kalman filter tuning with SML
4. Gray box model learning with SML
5. Discussion

MOTIVATIONS

Engineers need accurate models

- In many practical situations, engineers have to determine dynamical models of real systems from available data sets and prior knowledge.

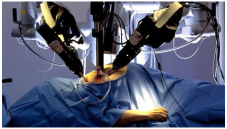
AIRBUS-ONERA-LIAS project

Fluttering detection
DDM from short duration data

GE-LAMIH-RU-LIAS project

Fouling detection in heat exchangers

SINTERS-ICUBE-LIAS project



Flexibility estimation and control for remote surgery

MICHELIN-LIAS project

Tire-road interaction estimation for autonomous cars

Université de Poitiers

Common denominator of these projects

- During these projects, we were asked to generate (new) estimators of physical parameters and signals.
  - "The best" parameter estimates were produced with nonlinear optimization based solutions.
  - "The best" signal estimates were produced with Kalman filters.
- Precise results were achieved solely through meticulous tuning of these techniques.
  - Nonlinear optimization works well when initial guesses are in the vicinity of the global optimum.
  - Kalman filters are efficient when the noise covariance matrices are well selected *a priori*.
- We primarily developed new tuning solutions based on initial estimates generated by subspace based model learning.

SUBSPACE BASED MODEL LEARNING: A REMINDER

Problem formulation

- Subspace based model learning (SML) methods mainly focus on state space models of the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{w}_k, \tag{1a}$$
$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{v}_k. \tag{1b}$$

where the noise sources are assumed to be realizations of zero mean white noises statically independent of the input sequence such that

$$\mathbb{E}\left[\begin{bmatrix}\mathbf{v}_i \\ \mathbf{w}_i\end{bmatrix}\begin{bmatrix}\mathbf{v}_j^\top & \mathbf{w}_j^\top\end{bmatrix}\right] = \begin{bmatrix}\boldsymbol{V} & \boldsymbol{S} \\ \boldsymbol{S}^\top & \boldsymbol{W}\end{bmatrix}\delta_{ij}. \tag{2}$$

Problem formulation (cont'd)

- By assuming that

  A1 the input vector sequence is quasi stationary and exciting of sufficient order,

  A2 the pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable and the pair $(\boldsymbol{A}, \begin{bmatrix} \boldsymbol{B} & \boldsymbol{V}^{1/2} \end{bmatrix})$ is reachable,

  standard SML solutions aim at estimating

  - the order n_x of the system,
  - an approximated minimum variance estimate of $\boldsymbol{x}_k$, $k \in \mathbb{T}$,
  - $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ up to a similarity transformation.

Main steps

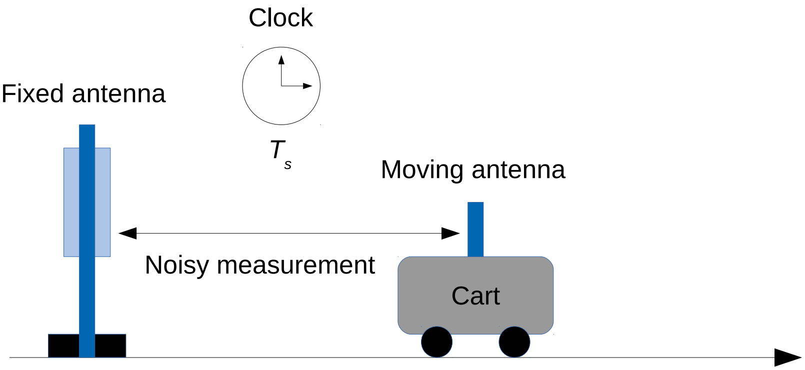
- Under Assumption A1 and A2, standard SML solutions consists in
  1. selecting future and past indexes f and p with the constraint that $f > n_x$ and p large "enough",
  2. building "past" and "future" Hankel matrices $\boldsymbol{Z}_p$, $\boldsymbol{U}_f$ and $\boldsymbol{Y}_f$,
  3. computing the RQ factorization of $\begin{bmatrix}\boldsymbol{U}_f^\top & \boldsymbol{Z}_p^\top & \boldsymbol{Y}_f\end{bmatrix}^\top$,
  4. extracting $\boldsymbol{R}_{32}$ and $\boldsymbol{R}_{22}$ from this RQ factorization, then computing the SVD of $\boldsymbol{R}_{32}\boldsymbol{R}_{22}^{-1}\boldsymbol{Z}_p$ to determine $\hat{n}_x$ as well as an estimate $\hat{\boldsymbol{X}}_{[f,M]}$ of the state on a user defined horizon,
  5. resorting to a LLS solution for determining $\hat{\boldsymbol{A}}$ $\hat{\boldsymbol{B}}$ and $\hat{\boldsymbol{C}}$.
- Keep in mind that all these estimates are valid up to a similarity transformation!!!!

KALMAN FILTER TUNING WITH SUBSPACE BASED MODEL LEARNING

KALMAN FILTERING: A REMINDER

Toy example

- Let us assume we want[1] to determine from remote noisy measurements the position and speed (state) of a cart moving straightforward.



[1]See *Understanding the basis of the Kalman filter via a simple and intuitive derivation*, R. Faragher, IEEE Signal Processing Magazine, 2012.

Main steps

- In order to reach this goal, we need[2]
  - a model of the cart dynamics, *i.e.*,

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k,$$

  - a model of the measuring process, *i.e.*,

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k,$$

[2] We focus on LTI systems only.

Main steps

- In order to reach this goal, we need
  - a model of the cart dynamics, *i.e.*,

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{w}_k,$$

  - a model of the measuring process, *i.e.*,

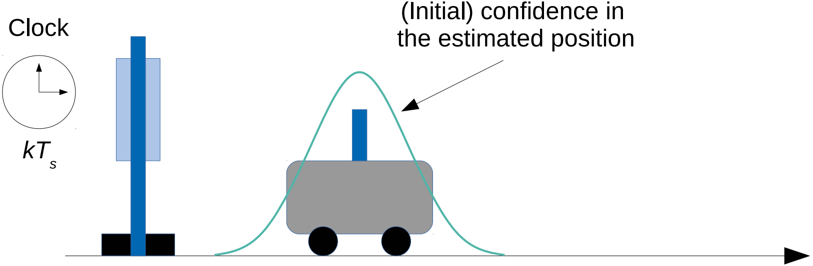
$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{v}_k,$$

  - a description of the noise and uncertainties acting on the system, *i.e.*,

$$\mathbb{E}\left\{\mathbf{w}_k\right\} = \mathbf{0},$$
$$\mathbb{E}\left\{\mathbf{w}_k\mathbf{w}_j^\top\right\} = \boldsymbol{W}_k\delta_{kj},\ \ \boldsymbol{W}_k \succ 0,$$

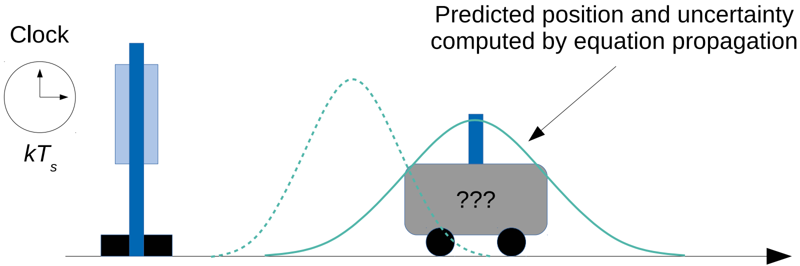
$$\mathbb{E}\left\{\mathbf{v}_k\right\} = \mathbf{0},$$
$$\mathbb{E}\left\{\mathbf{v}_k\mathbf{v}_j^\top\right\} = \boldsymbol{V}_k\delta_{kj},\ \ \boldsymbol{V}_k \succ 0.$$

Main steps (cont'd)



- Prior:
  - the model matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$,
  - the estimated state $\boldsymbol{x}_{k-1}^{+}$ and its covariance matrix $\boldsymbol{X}_{k-1}^{+}$.

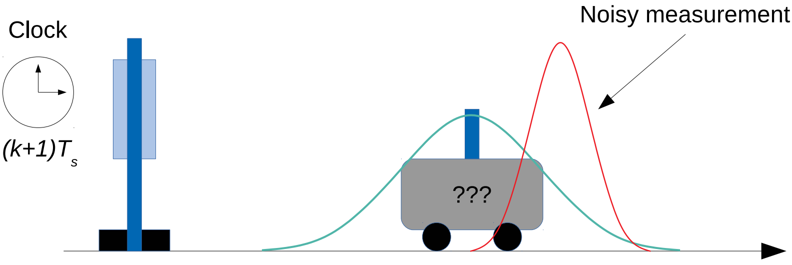
Main steps (cont'd)



- "Prediction" Kalman filter equations are

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1}^+ + \boldsymbol{B}\boldsymbol{u}_{k-1},$$
$$\boldsymbol{X}_k^- = \boldsymbol{A}\boldsymbol{X}_{k-1}^+\boldsymbol{A}^\top + \boldsymbol{W}_k.$$

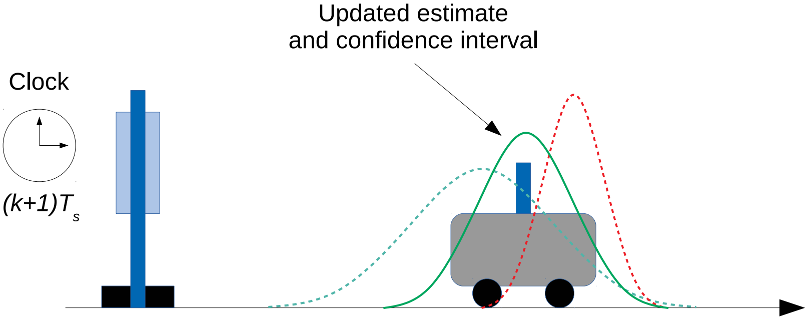
Main steps (cont'd)



- We get noisy measurements, *i.e.*,

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{v}_k.$$

Main steps (cont'd)



- "Update" Kalman filter equations are

$$\boldsymbol{K}_k = \boldsymbol{X}_k^- \boldsymbol{C}^\top \left(\boldsymbol{C} \boldsymbol{X}_k^- \boldsymbol{C}^\top + \boldsymbol{V}_k \right)^{-1},$$

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k (\boldsymbol{y}_k - \boldsymbol{C} \hat{\boldsymbol{x}}_k^-),$$

$$\boldsymbol{X}_k^+ = \left(\boldsymbol{I}_{n_x \times n_x} - \boldsymbol{K}_k \boldsymbol{C} \right) \boldsymbol{X}_k^-.$$

Basic idea of the solution

- The sequences $(\boldsymbol{v}_i)_{i \in \mathbb{T}}$ and $(\boldsymbol{w}_i)_{i \in \mathbb{T}}$ are used to describe
  - the noise acting on the real system,
  - the (in)accuracy of the model representation,

  thanks to their covariance matrices $\boldsymbol{V}_k$ and $\boldsymbol{W}_k$, $k \in \mathbb{T}$.
- Because $\boldsymbol{V}$ and $\boldsymbol{W}$ are used to describe the confidence we have in the model and the measurements, we aim at determining them by comparing
  - the model used in the Kalman filter,
  - a model estimated from the available data sets.
- Herein, the data driven model learning solution is a SML method.

Université de Poitiers

NOISE COVARIANCE MATRIX ESTIMATION

Main ingredients

- When a Kalman filter is designed, we have access to
  - I/O data samples,
  - a "reliable" DT state space representation of the system dynamics, *i.e.*, $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$.
- When the I/O data is rich enough, it can be used with a SML algorithm to generate
  - an approximated minimum variance estimate of $\boldsymbol{x}_k$, $k \in \mathbb{T}$,
  - estimates of $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$,

  up to a similarity transformation.

Main ingredients

- When a Kalman filter is designed, we have access to
  - I/O data samples,
  - a "reliable" DT state space representation of the system dynamics, *i.e.*, $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$.
- When the I/O data is rich enough, it can be used with a SML algorithm to generate
  - an approximated minimum variance estimate of $\boldsymbol{x}_k$, $k \in \mathbb{T}$,
  - estimates of $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$,

  up to a similarity transformation.

Notation interlude

For any vector $\boldsymbol{r}_k \in \mathbb{R}^{n_r \times 1}$ and parameters M, i and $\ell \in \mathbb{N}_*^+$, we define

$$\boldsymbol{R}_{[i,M]} = \begin{bmatrix} \boldsymbol{r}_i & \boldsymbol{r}_{i+1} \cdots & \boldsymbol{r}_{i+M-1} \end{bmatrix} \in \mathbb{R}^{n_r \times M}.$$

Similarity transformation

- The SML algorithm gives access to $\hat{\boldsymbol{X}}_{[i,M]}$ in an unknown basis.
- Knowing $(\boldsymbol{A}, \boldsymbol{C})$ and $(\hat{\boldsymbol{A}}, \hat{\boldsymbol{C}})$, the similarity transformation $\boldsymbol{T}$ between $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ and $(\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{C}})$ satisfies

$$\boldsymbol{\Gamma}_i(\boldsymbol{A}, \boldsymbol{C})\boldsymbol{T} = \boldsymbol{\Gamma}_i(\hat{\boldsymbol{A}}, \hat{\boldsymbol{C}}),$$

where

$$\boldsymbol{\Gamma}_i(\boldsymbol{A}, \boldsymbol{C}) = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{C}\boldsymbol{A} \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^{i-1} \end{bmatrix}, \; i \geq n_x.$$

- Thus, we can get $\breve{\boldsymbol{X}}_{[i,M]}$ is the "correct basis" as follows

$$\breve{\boldsymbol{X}}_{[i,M]} = \boldsymbol{\Gamma}_i^{\dagger}(\boldsymbol{A}, \boldsymbol{C})\boldsymbol{\Gamma}_i(\hat{\boldsymbol{A}}, \hat{\boldsymbol{C}})\hat{\boldsymbol{X}}_{[i,M]}.$$

Covariance matrix estimation

- We can finally estimate

$$\begin{bmatrix} \hat{\boldsymbol{W}}_{[i,M-1]} \\ \hat{\boldsymbol{V}}_{[i,M-1]} \end{bmatrix} = \begin{bmatrix} \breve{\boldsymbol{X}}_{[i+1,M]} \\ \boldsymbol{Y}_{[i,M-1]} \end{bmatrix} - \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \breve{\boldsymbol{X}}_{[i,M-1]} \\ \boldsymbol{U}_{[i,M-1]} \end{bmatrix},$$

and

$$\begin{bmatrix} \hat{\boldsymbol{V}} & \hat{\boldsymbol{S}} \\ \hat{\boldsymbol{S}}^\top & \hat{\boldsymbol{W}} \end{bmatrix} = \lim_{M\to\infty} \frac{1}{M} \begin{bmatrix} \hat{\boldsymbol{W}}_{[i,M-1]} \\ \hat{\boldsymbol{V}}_{[i,M-1]} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{W}}^\top_{[i,M-1]} & \hat{\boldsymbol{V}}^\top_{[i,M-1]} \end{bmatrix}.$$

NUMERICAL ILLUSTATIONS

Toy example

- We consider

$$\boldsymbol{A} = \begin{bmatrix} 0.603 & 0.603 & 0 & 0 \\ -0.603 & 0.603,0 & 0 & \\ 0 & 0 & -0.603 & -0.603 \\ 0 & 0 & 0.603 & -0.603 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 1.1650 & -0.6965 \\ 0.6268 & 1.6961 \\ 0.0751 & 0.0591 \\ 0.3516 & 1.7971 \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} 0.2641 & -1.4462 & 1.2460 & 0.5774 \\ 0.8717 & -0.7012 & -0.6390 & -0.3600 \end{bmatrix},$$

$$\boldsymbol{K} = 4 \times \begin{bmatrix} 0.1242 & -0.0895 \\ -0.0828 & -0.0128 \\ 0.0390 & -0.0968 \\ -0.0225 & 0.1459 \end{bmatrix},$$

$$\boldsymbol{R}_e = \begin{bmatrix} 0.0176 & -0.0267 \\ -0.0267 & 0.0497 \end{bmatrix}.$$

- We generate 10^3 realizations of the noise sequence and we select a data length $N = 1000$.

Estimates of the elements of $\boldsymbol{V}$

		$\hat{v}_{11}$	$\hat{v}_{12}$	$\hat{v}_{22}$
Theo. value		0.0176	-0.0267	0.0497
Sample cov.	avg.	0.0176	-0.0267	0.0497
	std.	0.0008	0.0012	0.0022
ICM	avg.	0.0588	-0.0750	0.0888
	std.	0.0412	0.0149	0.0057
DCM	avg.	0.031	-0.041	0.028
	std.	0.0041	0.0013	0.006
CMM	avg.	0.023	-0.073	0.031
	std.	0.0087	0.0066	0.0092
New meth.	avg.	0.0198	-0.0272	0.0516
	std.	0.0011	0.0013	0.0024

Estimates of the elements of $\boldsymbol{W}$

		$\hat{w}_{11}$	$\hat{w}_{12}$	$\hat{w}_{22}$	$\hat{w}_{23}$	$\hat{w}_{34}$	$\hat{w}_{44}$
Theo. value		0.0202	-0.0045	0.0149	-0.0198	0.0012	-0.0031
Sample cov.	avg.	0.0202	-0.0045	0.0149	-0.0198	0.0012	-0.0031
	std.	0.8886e-03	0.2054e-03	0.6563e-03	0.8750e-03	0.0509e-03	0.1487e-03
ICM	avg.	0.0526	-0.0150	0.0355	-0.0454	0.0041	-0.0103
	std.	0.0146	0.0079	0.0066	0.0067	0.0042	0.0036
DCM	avg.	0.0113	-0.0058	0.0186	-0.0285	0.003	-0.0103
	std.	0.004	0.0033	0.0068	0.0067	0.0039	0.0033
CMM	avg.	0.0170	-0.0041	0.0124	-0.0234	0.0041	-0.0043
	std.	0.0097	0.0082	0.0064	0.0062	0.0052	0.0028
New meth.	avg.	0.0196	-0.0041	0.0145	-0.0190	0.0015	-0.0026
	std.	0.0017	0.0006	0.0011	0.0011	0.0004	0.0005

Estimates of the elements of $\boldsymbol{S}$

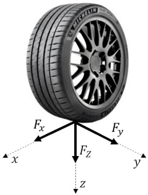
		$\hat{s}_{11}$	$\hat{s}_{13}$	$\hat{s}_{21}$	$\hat{s}_{24}$
Theo. value		0.0183	-0.0045	0.0131	-0.0172
Sample cov.	avg.	0.0183	-0.0045	0.0131	-0.0172
	std.	0.0008	0.0002	0.0006	0.0008
New meth.	avg.	0.0181	-0.0039	0.0137	-0.0169
	std.	0.0011	0.0007	0.0010	0.0010

MICHELIN PROJECT

PhD thesis with Michelin

- Future autonomous vehicles will require well-developed Advanced Driver Assistance Systems (ADAS) to assist human beings in driving.
- One path chosen by Michelin for ADAS improvement consists in providing ADAS with information related to the state of the road.
- Such information is included in the grip potential quantity.
- Benefits for passenger security (to name a few) are
  - detection of roads with low grip area,
  - evaluation of the driving conditions,
  - reduction of the impact of rear end collisions.

Problem formulation



- The grip potential is

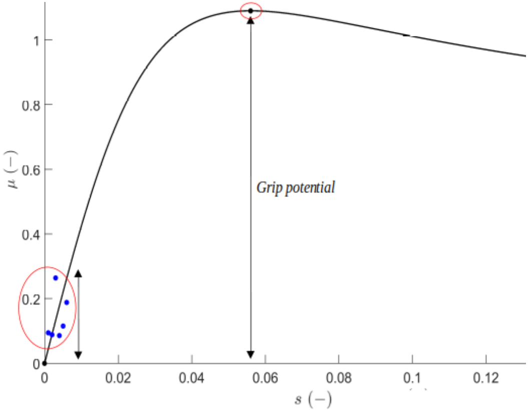
$$\mu_{\max} = \max\left(\frac{\sqrt{F_x^2 + F_y^2}}{F_z}\right),$$

i.e., the maximum effort a tire can generate before sliding on the road.

Problem formulation (cont'd)

Problem

Estimate the grip potential under standard driving conditions from sensors fitted on production vehicles.



Problem formulation (cont'd)

- Getting (noisy) data requires to measure the friction μ and the slip ratio s.
- No dedicated sensors exist on production vehicles.
- These signals must be estimated knowing that, for the longitudinal dynamics,

$$\mu = \frac{F_x}{F_z},$$
$$s = \frac{\omega R_{\mathsf{rol}} - v_x}{\max\left(\omega R_{\mathsf{rol}} - v_x\right)}.$$

- A Kalman filter was suggested to reconstruct the components of μ and s accurately.
- Efficient Kalman filter tuning solutions must be introduced to guarantee accurate estimates.

Université de Poitiers

Michelin results

- Data is generated with VI-CRT (realistic simulator).
- A nonlinear state space model is used, *i.e.*,

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

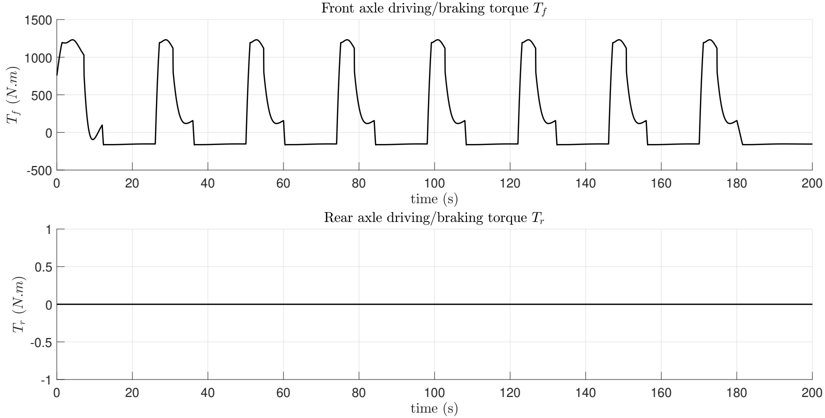
with

$$\boldsymbol{x} = \begin{bmatrix} v_x & \omega_f & \omega_r & F_{x_f} & F_{x_r} & \dot{F}_{x_f} & \dot{F}_{x_r} & \kappa & \dot{\kappa} \end{bmatrix}^\top,$$
$$\boldsymbol{u} = \begin{bmatrix} T_f & T_r \end{bmatrix}^\top,$$
$$\boldsymbol{y} = \begin{bmatrix} v_x & \omega_f & \omega_r & \dot{\kappa} \end{bmatrix}^\top,$$

involving

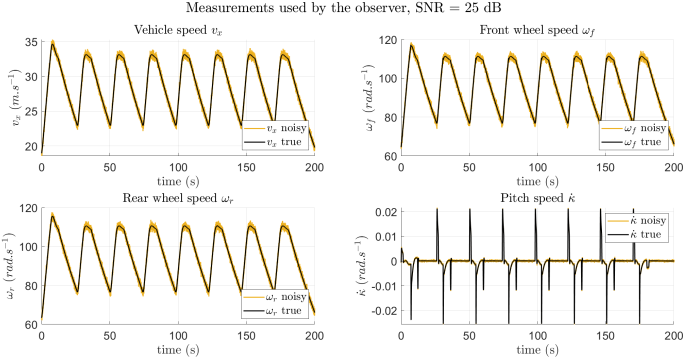
- a single track model for the vehicle dynamics,
- an effective tire radius model,
- a suspension model and a load transfer model.

Input data



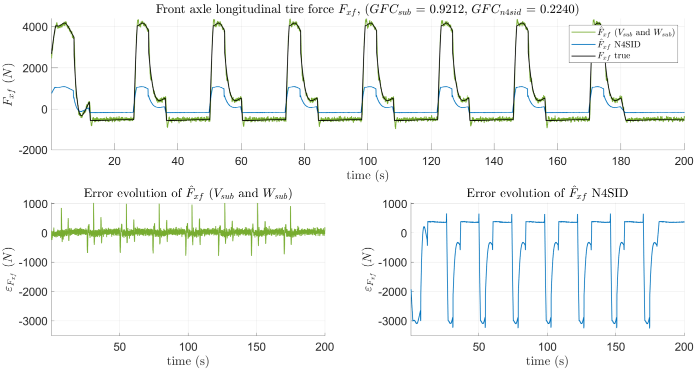
Université de Poitiers

Output data

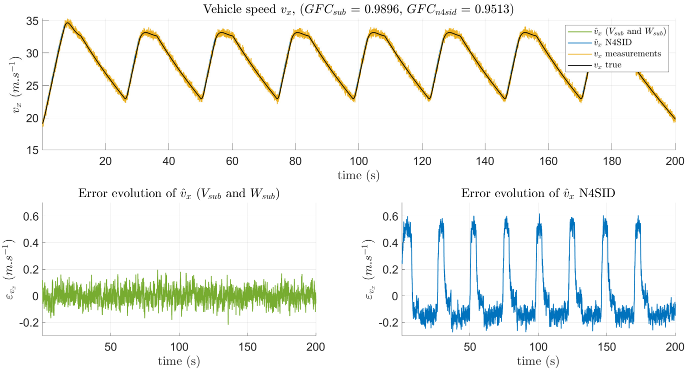


Measurements used by the observer, SNR = 25 dB
Vehicle speed v_x
v_x $(m.s^{-1})$
v_x noisy
v_x true
time (s)
Front wheel speed ω_f
ω_f $(rad.s^{-1})$
ω_f noisy
ω_f true
time (s)
Rear wheel speed ω_r
ω_r $(rad.s^{-1})$
ω_r noisy
ω_r true
time (s)
Pitch speed $\dot{\kappa}$
$\dot{\kappa}$ $(rad.s^{-1})$
$\dot{\kappa}$ noisy
$\dot{\kappa}$ true
time (s)

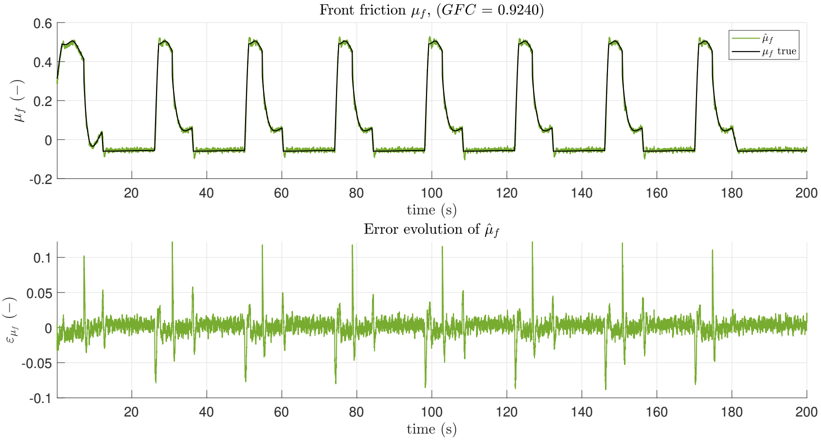
Long. tire force estimation



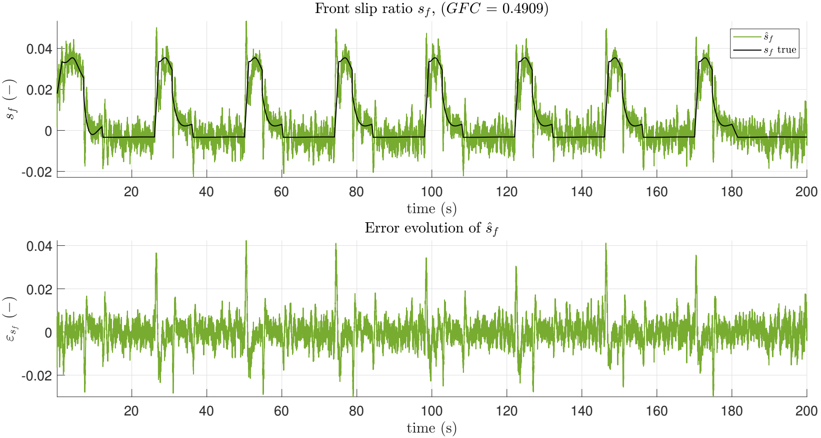
Vehicle speed estimation



Friction reconstruction

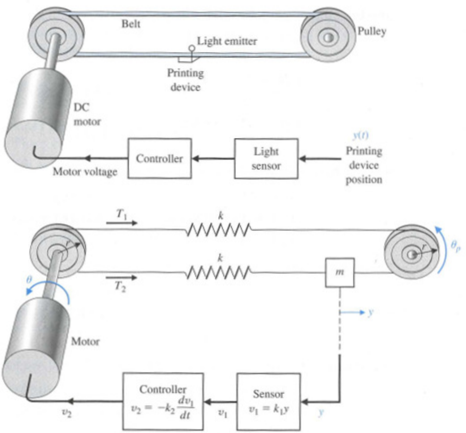


Slip ratio reconstruction



GRAY BOX MODEL LEARNING WITH SUBSPACE BASED MODEL LEARNING

Running example: a printer belt



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

Running example (cont'd)

- This system satisfies

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

with

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

and

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

Running example (cont'd)

- This system satisfies

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

with

$$\boldsymbol{A}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & -1 & \theta_1 \\ \theta_2 & 0 & 0 \\ \theta_3 & 0 & \theta_4 \end{bmatrix}, \qquad \boldsymbol{B}(\boldsymbol{\theta}) = \begin{bmatrix} 0 \\ 0 \\ \theta_5 \end{bmatrix},$$
$$\boldsymbol{C}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & \theta_6 & 0 \end{bmatrix}.$$

Running example (cont'd)

- This system satisfies

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

with

$$\theta_1 = r, \qquad \theta_2 = \frac{2k}{m}, \qquad \theta_3 = -\frac{2kr}{J},$$
$$\theta_4 = -\frac{b}{J}, \qquad \theta_5 = \frac{k_m}{RJ}, \qquad \theta_6 = k_1.$$

Problem

From the available I/O data, estimate consistently (and uniquely) a gray box (GB) model of this system.

A STANDARD SOLUTION

Output error method

- Fix a model parameterization

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

$$\boldsymbol{C}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & \theta_6 & 0 \end{bmatrix}, \qquad \boldsymbol{D}(\boldsymbol{\theta}) = 0.$$

Output error method

- Fix a model parameterization
- Check the identifiability

Structural identifiability

We must assume that k_1, m, r, and R are known *a priori* to guarantee that the model is structurally identifiable.

Output error method

- Fix a model parameterization
- Check the identifiability
- Compare with a signal norm the outputs of the system and the model, *e.g.*,

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

with

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

Output error method

- Fix a model parameterization
- Check the identifiability
- Compare with a signal norm the outputs of the system and the model, *e.g.*,

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

Remark

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

Simulation example

- Simulation procedure:
  - one noise free data set for model learning (PRBS)
  - A double Monte Carlo simulation is run

Simulation example

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

Simulation example

- Simulation procedure:
  - one noise free data set for model learning (PRBS)
  - A double Monte Carlo simulation is run
    - for the noise, 100 zero mean white Gaussian noises are generated such that the SNR $= 10\ dB$,
    - for the initialization, 100 initial guesses are generated as follows

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

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

Parameter estimation results

	NOMINAL	BEST	STD	MEAN	MEDIAN
k_1	1.4	1.4	0	1.4	1.4
m	0.2	0.2	0	0.2	0.2
r	0.15	0.15	0	0.15	0.15
R	2	2	0	2	2
b	0.25	0.24399	0.0030617	0.25072	0.25106
J	0.01	0.0091905	0.00034345	0.0099452	0.0099247
k	20	19.9461	0.051731	20.0234	20.0317
k_m	2	1.9554	0.024181	2.006	2.0088

Quick analysis

- Getting such good results require
  - reliable initial guesses,
  - tuning the nonlinear optimization efficiently,

  in order to help the algorithm converge towards the global minimum of V_N.

Some important questions

- Can we suggest alternatives to this output error approach?
- Can we generate reliable initial guesses for nonlinear optimization?

FROM BLACK BOX TO GRAY BOX WITH LINEAR ALGEBRA

Problem formulation

- Let use assume that
  - A reliable fully parameterized continuous[2] time linear time invariant 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.

[2] Several transformations exist to convert a DT LTI SS model into a CT LTI SS model efficiently.

Solution

- In [Prot and Mercère, 2020], we have shown that the similarity transformation $\boldsymbol{T}$ or $\boldsymbol{S}$ can be determined by
  - extracting a large system of equations linear in terms of the similarity transformation,
  - solving this system of linear equations by using standard linear algebra tools,
  - completing this two step procedure by a numerical optimization solution (dedicated now to a very small number of unknowns) if necessary,

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

A 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

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

A 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

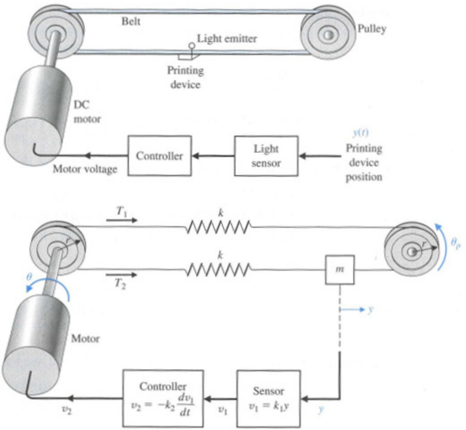
$$\begin{bmatrix} \mathfrak{A}^\top \otimes \boldsymbol{I}_{2\times 2} - \boldsymbol{I}_{2\times 2} \otimes \begin{bmatrix} 0 & 1 \end{bmatrix} \\ \boldsymbol{I}_{2\times 2} \otimes \begin{bmatrix} 1 & 0 \end{bmatrix} \\ \mathfrak{B}^\top \otimes \boldsymbol{I}_{2\times 2} \end{bmatrix} \text{vec}(\boldsymbol{S})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

A toy example (cont'd)

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

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

Printer belt (cont'd)



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

Printer belt (cont'd)

- We consider the model parameterization

$$\boldsymbol{A}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & -1 & \theta_1 \\ \theta_2 & 0 & 0 \\ \theta_3 & 0 & \theta_4 \end{bmatrix}, \qquad \boldsymbol{B}(\boldsymbol{\theta}) = \begin{bmatrix} 0 \\ 0 \\ \theta_5 \end{bmatrix},$$
$$\boldsymbol{C}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & \theta_6 & 0 \end{bmatrix}.$$

- We know (in red)

$$\boldsymbol{A}(\boldsymbol{\theta}) = \begin{bmatrix} \color{red}{0} & \color{red}{-1} & \color{red}{\theta_1} \\ \theta_2 & \color{red}{0} & \color{red}{0} \\ \theta_3 & \color{red}{0} & \theta_4 \end{bmatrix}, \qquad \boldsymbol{B}(\boldsymbol{\theta}) = \begin{bmatrix} \color{red}{0} \\ \color{red}{0} \\ \theta_5 \end{bmatrix},$$
$$\boldsymbol{C}(\boldsymbol{\theta}) = \begin{bmatrix} \color{red}{0} & \color{red}{\theta_6} & \color{red}{0} \end{bmatrix}.$$

Université de Poitiers

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

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

Printer belt (cont'd)

- By using the vectorization tool

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

- 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.

Université de Poitiers

Printer belt (cont'd)

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

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

  where $\boldsymbol{\alpha}_{\mathsf{null}}$ 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

Printer belt (cont'd)

- We introduce the cost function

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

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

- 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 N4SID + d2c.
- We select the best black box model $(\mathfrak{A}, \mathfrak{B}, \mathfrak{C})$ giving the best fit on a validation data set.
- The initialization of the BFGS algorithm is performed with 100 different initial vectors $\boldsymbol{z}$ randomly generated from a standard zero-mean normal distribution.
- The best $\boldsymbol{z}$ is chosen by selecting the $\hat{S}$ having the best condition number.

Université de Poitiers

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
k_1	1.4	1.4
m	0.2	0.2
r	0.15	0.15
R	2	2
b	0.25	0.24401
J	0.01	0.0091916
k	20	19.9462
k_m	2	1.9555

Université de Poitiers

Explanations with the printer belt (cont'd)

- Let us assume that k 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

$$\underbrace{\begin{bmatrix} \mathfrak{A}^\top \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} - \boldsymbol{I}_{3\times3} \otimes \begin{bmatrix} 0 & -1 & \theta_1 \\ \theta_2 & 0 & 0 \end{bmatrix} \\ \boldsymbol{I}_{3\times3} \otimes \boldsymbol{C}(\boldsymbol{\theta}) \end{bmatrix}}_{\boldsymbol{N}}$$

$$\times \text{vec}(\boldsymbol{S}) = \underbrace{\begin{bmatrix} \mathbf{0}_{6\times1} \\ \text{vec}(\mathfrak{C}) \end{bmatrix}}_{\boldsymbol{\alpha}}.$$

- Now $\boldsymbol{N} \in \boldsymbol{R}^{9\times9}$ 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
k_1	1.4	1.4
m	0.2	0.2
r	0.15	0.15
R	2	2
b	0.25	0.25483
J	0.01	0.0099987
k	20	20
k_m	2	2.0367

Université de Poitiers

DISCUSSION

References I

Mussot, V., Mercère, G., Dairay, T., Arvis, V., and Vayssettes, J. (2021).
Noise covariance matrix estimation with subspace model identification for Kalman filtering.
International Journal of Adaptive Control and Signal Processing, 35:591–611.

Prot, O. and Mercère, G. (2020).
Combining linear algebra and numerical optimization for gray-box affine state-space model identification.
IEEE Transactions on Automatic Control, 65:3272–3285.