

# Jury members' comments on the private PhD defense of Gustavo Quintana-Carapia

Thank you to the jury members for reviewing the thesis manuscript and for giving suggestions for improvement. The manuscript changes due to the comments are highlighted in [blue fontcolor](#).

## General comments

- The manuscript text should better position the PhD to related work. Improve the literature review and discuss (and cite) the state of the art.

**The introduction of the chapters now includes the state of the art to position the PhD work with respect to the literature.**

- Add more clearly the motivation for the PhD research work by including motivational examples to the introduction.

**Motivational examples have been added to the Introduction.**

- Add a discussion with comparison of existing EIV methods. Give, with the positive and negative aspects in mind, a better justification for why you use a (R)LS method, and not, e.g., (R)IV method.

**In Section 4.1.4, a discussion that compares IV methods with EIV methods has been added.**

- Discuss more clearly the aspects of scalability of the methods and their limitations in terms of model order, size of data sets (e.g., related to the inversion of LS matrices).

**In Section 3.2 was added a discussion on the scalability of the data-driven step input estimation method.**

- The equations on pages 12 and 13 seemingly contain errors and should be corrected.

**The equations on pages 12 and 13 have been corrected.**

- The figure shown on page 66 was made with a single experiment; it is desirable to see the (averaged) effect on several experiments.

**A series of 100 runs of the maximum-likelihood estimation method has been simulated and the results have been added to the manuscript in Section 6.3.2.**

## Lyudmila Mihaylova

- You develop results for uncertainty quantification; what type of uncertainties can your approach deal with and which levels of noise can you consider? Elaborate on SNR levels; I'm also very interested in scalability, so how big the matrices can be.

A numerical sensitivity analysis was conducted on both subspace and maximum-likelihood methods for solving the affine input estimation problem. The sensitivity analysis showed that the subspace method can handle uncertainty on the generating system parameters up to 5% without affecting deeply the performance of the method. Moreover, the mean-squared error of the estimation computed by the subspace method is similar in up to one order of magnitude in the SNR interval from 20 to 60 dB, being  $10^{-1} \text{ g}^2$  its maximum value at 40 dB. The levels of measurement noise in practical applications are found in this SNR interval.

- Concerning the scalability to data: eqn 3.6: how scalable is the approach in all aspects (order, data, linked to inversion for LS matrix); can you deal with hundreds, thousands of data?

The following paragraph has been added in Section 3.2 to discuss the scalability of the data-driven step input estimation method.

The computational complexity of the RLS algorithm solution to the system of equations (3.14) is  $O((n+1)^2)$ . The largest computational requirement is for the initialization of the algorithm, where a matrix inverse is required when the number of samples is just enough to have a square matrix  $\tilde{\mathbf{K}}_{n+1}$ . From there on, the RLS algorithm updates the solution with linear complexity. Therefore, even though a large number of samples  $N$  makes the matrix in the system of equations (3.14) increase in the number of rows, the estimation of the solution does not increase in complexity. The data-driven step input estimation method is then scalable for any sensor of order  $n$ , and can be executed in devices with limited computational resources, provided that the computation of the  $n+1 \times n+1$  inverse matrix is feasible.

- What recommendation up to scale/order can method work with. What are the limitations? It would be good to have more precise discussion about these aspects in dissertation: what do you mean by large scalability: 100,  $10^4$ , ...? A more detailed discussion is desirable.

In Section 3.2 was added a discussion on the scalability of the data-driven step input estimation method. Since it uses recursive least squares to update the estimation, its complexity is linear in the number of parameters  $n+1$ . This means that for small number of samples,  $N = 100$ , or large number of samples,  $N = 10^4$ , the complexity per recursion does not change.

The computation of the bias and variance of the input estimation, as it is proposed in Chapter 4, has a complexity  $O(n^3N)$ . This complexity prevents the implementation in real time of these statistical moments estimation. A future research work can be devoted to find efficient estimation of these input estimation moments, to be obtained right after the input estimation itself.

- On slide 25 (noisy graphs) it would be good to see a statistical analysis (more experiments than one).

100 runs of the maximum-likelihood estimation method have been simulated and their results have been added to the manuscript in Section 6.3.2.

The maximum-likelihood (ML) method estimated the affine input parameters from the sensor model transient response. A total of 100 runs of the method were obtained, using different realizations of measurement noise. The measurement noise variance was set to have an SNR of 40 dB. The ML method used the first 50 samples to initialize the optimization variables and updated the variables every  $N_s = 5$  samples. Figure 6.8 shows the average of the observed relative errors in the estimation of the parameters  $\hat{a}$ ,  $\hat{b}$ ,  $\hat{x}_{\text{ini},1}$ , and  $\hat{x}_{\text{ini},2}$ .

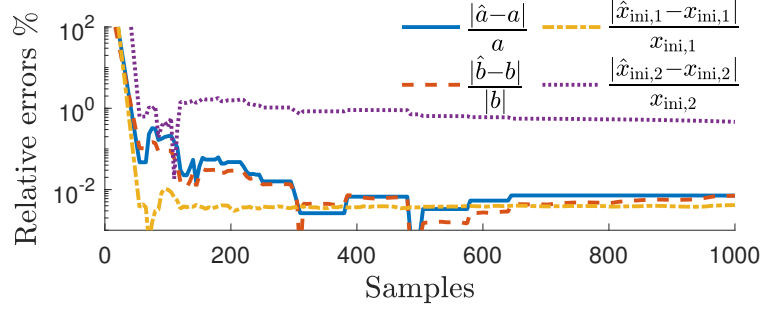


Figure 6.8: The affine input parameters and the sensor's initial conditions are estimated with the ML method. After three iterations, the average of the relative error is smaller than 5 % for each and every of the estimates. The estimate  $\hat{x}_{ini,2}$  has the larger relative error near to 1 %.

Figure 6.9 shows the average of the variances that were observed on the diagonal of the covariance matrix  $\mathbf{J}$ . We can see that the estimation variances decrease as more samples are processed. Moreover, the estimation variances of  $\hat{a}$  and  $\hat{b}$  obtained with the ML method are lower than the corresponding estimation MSE errors obtained from a Monte Carlo simulation of the subspace method.

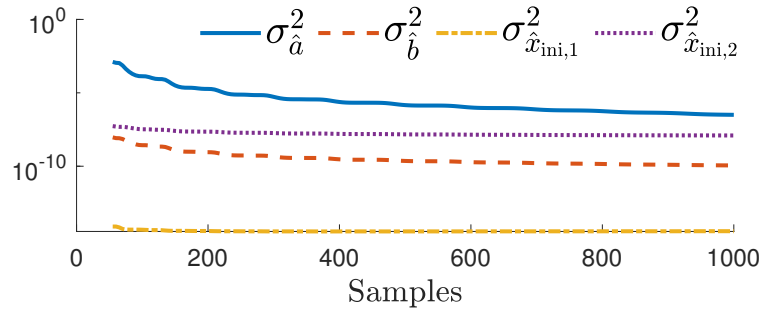


Figure 6.9: The variances of the ML estimates are calculated using the information provided by the analytic Jacobian. The estimates  $\hat{x}_{ini,1}$  and  $\hat{a}$  have the smaller and larger variances, respectively.

## Stephane Chretien

- The Taylor expansion is computed of order 2, what is the accuracy of expansion with respect to system, noise level,... Can you come up with a priori bound on error that will allow you to assess the quality of the expansion? Can you predict where the solution is going to be before computing Taylor expansion?

**An a priori bound on the error introduced by the Taylor series expansion can be obtained in terms of the spectral radius of the matrix  $\mathbf{M}$ . In Section 4.1 the following description has been added:**

An a priori bound on the error introduced by the second order Taylor series expansion can be expressed in terms of the spectral radius  $\rho(\mathbf{M})$ . Considering that the Taylor series expansion

is an infinite sum we have that

$$\begin{aligned}
(\mathbf{I} + \mathbf{M})^{-1} &= \mathbf{I} - \mathbf{M} + \mathbf{M}^2 - \mathbf{M}^3 + \dots \\
&= \mathbf{I} - \mathbf{M} + \mathbf{M}^2 - \mathbf{M}^3 (\mathbf{I} - \mathbf{M} + \mathbf{M}^2 - \mathbf{M}^3 + \dots) \\
&= \mathbf{I} - \mathbf{M} + \mathbf{M}^2 - \mathbf{M}^3 (\mathbf{I} + \mathbf{M})^{-1}
\end{aligned} \tag{4.4}$$

Therefore we can obtain a bound for the second order expansion error if we take a matrix norm as follows

$$\begin{aligned}
\|(\mathbf{I} + \mathbf{M})^{-1} - (\mathbf{I} - \mathbf{M} + \mathbf{M}^2)\| &= \|\mathbf{M}^3(\mathbf{I} + \mathbf{M})^{-1}\| \\
&\leq \|\mathbf{M}^3\| \|(\mathbf{I} + \mathbf{M})^{-1}\| \leq \frac{\|\mathbf{M}\|^3}{1 - \|\mathbf{M}\|} = \frac{\rho(\mathbf{M})^3}{1 - \rho(\mathbf{M})}
\end{aligned} \tag{4.5}$$

- Concerning other sorts of prior estimations, e.g., given by explicit function theorems. I wonder if you could get something and apply this bound in order to control the bound of the 2nd order expansion. There exist some good results that could help you devise something a priori which can control the error you are making. Neuberger's results (quite old now) but may be handy for this kind of problem. This can be investigated further.

**The investigation of the error bounds for the second order Taylor expansion can be conducted using the continuous Newton method. This is proposed as future research work in Chapter 7.**

Neuberger's paper [1] introduces five theorems that are consequence of the continuous Newton method. The Newton method aims to find a zero of the function  $F : R^n \rightarrow R^n$  by iterating

$$z_{k+1} = z_k - (F'(z_k))^{-1} F(z_k) \tag{1}$$

for  $k = 0, 1, 2, \dots$ , and for a given initial  $z_0$ , assuming that  $(F'(y))^{-1}$  exists for some  $y \in R^n$ . A domain of attraction is the set of all initial values  $z_0$  that lead to a root of  $F$  after the convergence  $z_0, z_1, z_2, \dots$ . The damped Newton method

$$z_{k+1} = z_k - \delta_k (F'(z_k))^{-1} F(z_k) \tag{2}$$

prevents convergence issues, such as chaotic domains of attraction, with  $\delta_1, \delta_2, \dots \in (0, 1)$ . The continuous Newton method is a sequence of damped Newton methods. To implement the continuous Newton method, select  $0 < T < 1$  and perform  $m$  runs of the damped Newton method with  $\delta_k = T/m$ , for  $k = 1, 2, \dots, m$ . The result  $z_{k+1}$  is the zero estimation  $x_m$ , and if the sequence of estimates  $x_1, x_2, \dots$  converges, then it converges to the result of the continuous Newton method.

The objective of the continuous Newton method is to find a function  $z : [0, \infty) \rightarrow R^n$ , so that

$$z(0) = x \in R^n, \quad z'(t) = - (F'(z(t)))^{-1} F(z(t)), \quad t \geq 0, \tag{3}$$

subject to the existence of the limit  $u = \lim_{t \rightarrow \infty} z(t)$ , that satisfies  $F(u) = 0$ . This implies

$$F'(z(t))z'(t) = -F(z(t)), \tag{4}$$

that is equivalent to

$$(F \circ z)'(t) = -F(z(t)), \tag{5}$$

from where we get

$$F(z(t)) = e^{-t}F(z(0)). \quad (6)$$

Thus, the residual  $F(z(t))$  only changes in magnitude and not in direction. This property is essential for the results of the theorems in [1]. In particular, Theorem 5 gives the conditions for the continuous Newton method to ensure finding a root of the function.

**Theorem 5** *Suppose that each of  $H$ ,  $J$ , and  $K$  is a Banach space, with  $H$  compactly embedded in  $J$ , that  $r > 0$ , and that  $G : B_r(0) \rightarrow K$  is continuous as a function on  $J$ . Suppose also that  $g$  belongs to  $K$  and that for each  $y$  in  $B_r(0)$  there is an  $h$  in  $B_r(0)$ , where  $b_r(0)$  and  $B_r(0)$  are open and closed balls in  $H$  of radius  $r$  centered at 0, such that*

$$\lim_{t \rightarrow 0+} -\frac{1}{t} (G(y + th) - G(y)) = g. \quad (7)$$

*Then, there exists  $u$  in  $B_r(0)$  such that  $G(u) = g$ .*

The theorem implies that  $u$  is a zero of the function defined as  $F(y) = G(y) - g$ . Since  $G$  is a continuous function and if  $(G'(y))^{-1}$  exists for each  $y$  in  $B_r(0)$ , a metric for the validity of the Theorem (5) hypothesis is the inequality  $\|(G'(y))^{-1}g\| \leq r$ .

To link the data-driven step-input estimation problem with this theorem, we can take the system of linear equations  $\tilde{\mathbf{y}} = \tilde{\mathbf{K}}\boldsymbol{\theta}$ , then define  $G(y) = \tilde{\mathbf{K}}\boldsymbol{\theta}$ , and  $g = \tilde{\mathbf{y}}$ , so that we have  $F(\boldsymbol{\theta}) = G(\boldsymbol{\theta}) - g$ . The derivative of the function  $G(\boldsymbol{\theta})$  is given by  $G'(\boldsymbol{\theta}) = \tilde{\mathbf{K}}\mathbf{I}$ . Using the pseudo-inverse of  $\tilde{\mathbf{K}}$ , the metric  $\|\tilde{\mathbf{K}}^\dagger \tilde{\mathbf{y}}\| \leq r$  indicates that the estimate  $\hat{\boldsymbol{\theta}}$  is found inside a ball of radius  $r$ . The questions that remain open are how large can  $r$  be?, and how is the radius  $r$  related to the perturbation noise variance?

- About trying to find a good transformation from LS to EIV model. I am wondering if you can think of other more accurate starting points instead of LS. Closer to the problem you want to study, but still preserving computational lightness? The closer you start from objective, better the Taylor expansion would be.. Something to investigate as well...

To get starting points to the EIV model, that are more accurate than LS, we would need to incorporate more prior knowledge. In the formulation of the EIV problem we have made assumptions of the model class for the sensor and the Gaussian white noise of given variance for the additive measurement noise. The LS is an heuristic algorithm that was chosen by its simplicity for solving online the estimation problem by means of recursion. However, the LS solution is biased. The optimal solution could be obtained with a maximum likelihood algorithm provided we have a large amount of samples, but we need to provide approximations starting from small number of samples. Moreover, the maximum likelihood solution of the EIV problem needs structured total least squares, which does not have a closed form that could be extended with a Taylor series, or structured total maximum likelihood, which do not support the correlation between the regression matrix and the regressor in the EIV problem. The recursive versions of these two structured approaches is also more computational expensive than RLS. Therefore, it is required to continue the research towards efficient solution algorithms.

- About robustness: is it an issue in practical problems with heavy tailed noise, outliers,... Perhaps the use of median of means plugged into your approach might produce interesting alternatives for outliers or heavy tailed data. But this is an intricate thing to deal with. Might be interesting to ponder this in future.

There are some estimators that can deal with heavy tailed noise. In the paper [2] there are described different estimators for a linear regression  $Y_i = b + aX_i + Y_i^*$ ,  $i = 1, \dots, n$ , where both the explanatory variables  $X_i$  and the errors  $Y_i^*$  have heavy tails. The  $X_i$  are samples from a Pareto distribution on  $(1, \infty)$  arranged in decreasing order  $X_n < \dots < X_2 < X_1$ , as if they were generated from the order statistics  $U_1 < \dots < U_n$  of a sample of uniform variables in  $(0, 1)$  by setting  $X_i = 1/U_i^\zeta$ . The parameter  $\zeta$  is the tail index of the Pareto distribution. For the variables  $Y_i^*$ , the tail index is  $\eta$ . Heavy tails are indicated by larger tail indexes. For  $\eta < 1/2$ , the variable  $Y^*$  has finite variance and least squares is the suggested estimator. For  $\zeta < 1/2$ , the variable  $X$  has finite variance and the least absolute deviation is the suggested estimator, otherwise the Theil-Sen estimator is preferable. The paper presents empirical results comparing the estimation bias and standard deviation for different tail indexes.

Regarding outliers, in [3] is described an approach of using least squares to identify the outliers, followed by a recalculation without considering the outliers. In this sense, one could also consider that the removed outliers are missing data, and use low-rank approximation. The classical approach is using a maximum likelihood type estimator (M-estimators) as it is described in [4] and [5]. The M-estimators aim to mitigate the outliers impact by using a function of the residual  $\rho(r_i)$  instead of the squared residuals  $r_i^2$  in the minimization  $\min \sum \rho(r_i)$ . The function  $\rho$  is symmetric, positive definite, with only one minimum at 0, and increasing slowly than the square.

Selecting another estimator to solve the EIV problem constructed by the data-driven step input estimation problem to deal with heavy tailed noise, or with outliers, might have a big impact on the implementation. The implementation may be computationally complex and not suitable for online applications.

- On deep learning: could this be an alternative?

Deep learning could be used but it would be an overkill methodology for linear estimation. However, for metrology applications where the input is the to-be-estimated signal, implementing deep learning could be a challenge. The neural networks are suitable for non-linear model based applications where the training requires the design and application of input stimuli to correlate with the corresponding outputs of the system. In metrology, it is not possible to train deep learning methodologies with real input data, because the input data is unknown and cannot be observed. By applying neural networks trained by simulation would estimate inaccurately the desired inputs.

## Guillaume Mercere

- In the first or second sentence: “users cannot wait for a steady state regime” . Do you have specific examples of real industrial systems for which it is necessary to have continuous read-

out before steady-state is reached? - Further on the industrial part and selling sensors with data-driven parts. Can you explain to experts that your solution is better? Sensors will be used for ages; why not spend one day to do system identification and get a reliable model to make a compensator, instead of using your approach?

**The fast weighing of parcels in the post office is a typical example of the real life systems where the time is limited to perform the measurements. Another one that we see nowadays is the temperature measurement of people in airports or train stations to prevent the spread of corona virus. The data driven method can be implemented in portable devices with limited computational resources. In industrial plants where a time slot can be devoted to identify the measurement system and implement a compensation method, the solution might be useful only for a certain amount of time. It is assumed that the model approximation deviates due to changes in the system elements, leading to inaccuracies in the input estimation. There is a need for adaptive methods to avoid stopping the production lines to repeat the system identification procedure. The data-driven method is such an adaptive method.**

- Chapter 2, Section 2.2.1. Can you give me some classic solutions/algorithms instead of LS in general? (Say I'm not familiar with EIV or TLS?) Usually with LS problems: multiply with  $K^T$  and compute pseudoinverse. Do you know another solution where they don't use  $K$  on the left-hand-side but something else? To get rid of noise by using a matrix on the left-hand-side, is related to EIV. There are some recursive solutions for this. It could be interesting to compare with such IV solutions.

**In Section 2.2.1 the following paragraphs have been modified to include references of instrumental variable solutions to the errors-in-variables problems.**

Instead of using a total least-squares (TLS), or an instrumental variables, solution of the structured EIV problem, the step input estimation method uses the recursive least-squares solution (RLS). In the review conducted by [6] it is shown that the TLS solution of unstructured EIV problems is consistent when the perturbations have zero mean with a given positive definite covariance, and the TLS solution is equivalent to the maximum likelihood (ML) solution when the disturbances of the EIV problems are i.i.d. normally distributed. The classical TLS solution is obtained using singular value decomposition of the augmented matrix built from the regression matrix and the regressor as additional columns. However, the studies presented in [7] demonstrate that the structured TLS solutions cannot be generalized since each structure in an EIV problem requires a specific treatment. Moreover, the ML estimator of structured EIV problems leads to non-convex optimization, and the global optimum is not guaranteed [8]. On the other hand, the instrumental variable (IV) solution of EIV problems use an additional variable. The additional variable is an instrument with the purpose of improving the consistency of the estimate. To do this, the construction of the instrumental variable has to be done in such a way that it is uncorrelated with the perturbations and correlated with the regressor, as it is described in [9]. Unfortunately, the computational complexity of the TLS, ML, and IV solutions inhibits their real-time implementation.

The solution of the step input estimation problem is preferred to be online, since its applications are in the field of metrology. In the literature there are references for online implementations of TLS, ML or IV methods, which are mainly recursive versions of the solution methods. These recursive versions are developed to solve EIV problems for system identification and fault detection. We have, for example, generalized recursive TLS solutions

introduced in [10] and [11], and recursive IV solutions developed in [12], [13] and [14]. These methods cannot be used as off-the-shelf solutions to the step input estimation EIV problem because they require to be adapted to the specific requirements of the problem. One is that the sensor output is the only measured signal, since the input is to-be estimated, and the solution should be simpler, preferable with a linear complexity  $O(n)$ , instead of the cubic  $O(n^3)$  or quadratic  $O(n^2)$  complexities of the mentioned recursive methods, where  $n$  is the sensor order. The proposal of the data-driven step input estimation method includes the recursive least-squares (RLS) solution. Compared to recursive TLS or recursive IV solutions, RLS is suboptimal but more simple solution to the structured EIV problem with lower computational complexity, suitable for real-time implementation.

- In Chapter 3: at beginning you say to use linear systems theory to model a sensor. Can you discuss this assumption: is it true in practice? Can you guarantee it behaves like an LTI system?

**The first paragraph of Chapter 3 has been added to explain this concept:**

The dynamic systems in the real world are non linear. Under certain intervals of operation, the linear approximations can explain the dynamics of the systems. The linear and time invariant systems have been studied in depth and the linear theory is a solid collection of tools to analyze and explain the behaviour of dynamic systems in a wide range of applications. On the other hand, the study of non-linear systems is more complex and strongly depends on the type of non-linearity that each particular systems exhibit. When the non-linear modeling does not apport a big amount of benefits into the application study, such as significative accuracy improvement or supporting wide ranges of operation, then is better to resort to the utilization of LTI representations. A sensor is a dynamic system whose operation range is preferred to be linear, to facilitate the repeatability and reproducibility of the measurements. This means that every time the sensor must deliver identical responses for the same input excitations.

- Equation 3.1: an LTI system, described with SS representation. You only put output noise on this representation. People sometimes also add process noise. Why? How to modify your approach if there is process noise? Is it still white in this case?

**In Chapter 3 was added the following paragraph to explain the impact of white process noise in the response of al LTI system.**

The process noise is the representation of the uncertainty in the system state due to un-modeled perturbations that affect the state evolution. The process noise is modeled as an additional input to the system excitation. For a linear time invariant (LTI) system, the response to a sum of two input excitations is the sum of the two responses that correspond to the individual inputs. It is demonstrated in [15] and [16] that the response of an LTI system to a white noise input is colored noise, since the power spectral density of the response is not constant, but instead proportional to the squared magnitude of the system impulse response. Nevertheless, in the bandwidth operation of a properly designed measurement system, processing white noise results in approximately white noise output. Moreover, in the present work the measurement noise exists only in the system response signal, because the input is not directly observed, and the sensor operation is not subject to external perturbation. Under these conditions, the mean and variance of the measurement noise  $\epsilon$  describe implicitly the process noise entering the system.



- On page 12 standard approach top of page equation: there are some mistakes in the equations! there is no epsilon (noise) in the equation. But then you do an SVD of  $H$  and estimate from there. How does it work if you have noise? Have you tried implementing it and see what you get?

The step input estimation based on singular value decomposition works only with exact data. When noise perturbs the sensor response the method cannot guarantee the step input estimation. The singular value decomposition of a Hankel matrix constructed from a perturbed sensor response provides only an approximation of the singular values and the singular vectors of this Hankel matrix. Then, the step input estimation obtained with this method from perturbed sensor response is only a rough estimation.

- On page 12: I'm not sure the equations at the end of the page are correct, e.g.,  $Y=G \bar{u}$  ... not sure there is  $x$ ; same for rest; There seem to be many mistakes here and next pages. Be careful with the equations!

The equations have been reviewed and the typos were corrected.

- On page 13, you assume persistency of excitation of order  $L$ . How do you check this? In your case the input is constant; how to guarantee that the assumption is met?

The persistency of excitation of a signal is verified by looking at the rank of the Hankel matrix constructed from the consecutive samples of this signal. When the Hankel matrix of  $L+1$  block rows has a rank lower or equal than  $L$ , then we say that the signal is persistently exciting of order  $L$ . A step input has a very low order of persistency  $L=1$ , but the step response of a system has a larger order of persistency. In fact, we assume that if the persistency of excitation of the step response is sufficiently high, then the columns of the Hankel matrix constructed from the step response span a linear space with all the responses of the equivalent augmented autonomous system where the step input is considered an augmented initial condition.

- In Chapter 4, p 17, Section 4.1.1: You assume that perturbations are IID and must be normally distributed. Why using an assumption of normally distributed? I would think that the kind of distribution is important for CRLB, but I'm not sure you need it for eqns 4.8, 4.9, 4.10,... IID is important, normality perhaps not.

The following paragraph has been added in Section 4.1 to discuss on the need for the normality assumption.

The normality assumption is necessary to provide more prior information into the method formulation. The second order Taylor series expansion (4.6) is developed considering the perturbation noise that gets into the regression matrix from the sensor transient response. Assuming only that the perturbation noise is distributed with zero mean and variance  $\sigma_\epsilon^2$  is not enough. If we additionally assume normality for the perturbation noise, then we benefit from the knowledge of the third moment equal to zero due to symmetry, and the fourth moment equal to three times the squared variance.

- In equation 4.11 you have  $\sigma_e p s^2 + \sigma_e^2$ ; how to get in practice  $\sigma_e$ ? Do you have to do this each time? Recursively?

The measurement noise variance is prior information and in Chapter 5 we describe two ways of estimating  $\sigma_\epsilon$ . In summary, the measurement noise can be

extracted from the sensor response once the transient effects are smaller than the noise floor, or from the sensor response power spectrum. The online estimation of the measurement noise, simultaneously with the online step input estimation is a topic for future research.

- Switch to 4.2, simulation results: first part dedicated to  $K$ , randomly generated; not sure if still possible, but in all your results it would be nice to have somewhere explicitly  $K = \dots$  (even if randomly generated). You use only one realization also for matrices  $A$ ,  $B$ ,  $C$ ,  $D$  in the text, so it can be added explicitly.

Section 4.2 has been improved to explain how the exact data and the perturbation noise are generated. This modification allows to reproduce the simulations conducted and understand the outcome of them.

- in Fig 4.1 the bottom figure also discusses the structured case. Why do you talk about structured results in a section about the unstructured case?

Section 4.2 has been improved to describe the result next to its discussion.

- In 4.1 uses a constant  $K$ , but in 4.2.2 you use  $A$ ,  $B$ ,  $C$ ,  $D$ , without any links to  $K$ . You compare the two curves generated from different data sets... It is strange to compare them. I don't see the link between both.

Section 4.2 has been restructured and the simulation for the unstructured case were redone to establish a link between the two cases. The exact data is the same in the two cases and the perturbation noise follows the corresponding rules of each case, with the same variance noise levels. In this way the comparison between the two cases makes more sense.

- To show comparison between structured and unstructured case, use unstructured results for structured case; then use unstructured results for structured case and cover all the possibilities. Now you compare curves related to two different simulations.

The numerical study of the expressions that calculate the step input estimation bias and variance has been appended to include the results of using unstructured perturbation in the expressions for the structured case, and also structured perturbation in the expressions for the unstructured case. It is expected that the performance is very far from that obtained when using the expressions in concordance with the type data they were derived for. A discussion on this observations is included in Section 4.2.3.

- I was surprised to read in Chapter 5 something very close to what I read in chapter 4. In both cases generated  $A$ ,  $B$ ,  $C$ . In 5.1 again oscillating 5th order system, similar conclusions, ... What is the added value of 5.1 wrt 4.2.2?

The derivation described in Chapter 4 was conducted with a second order mass-spring-damper weighing system example on mind, and the results observed were submitted in the corresponding paper. Afterwards, when we moved to the practical implementation, we faced the increase of the model order after setting up the weighing experiment. The lowest order we can assume is fifth order. So we decided to repeat the simulations and these new results serve two purposes: a transition from simulation to experimental results, and also to get an insight on the impact of the bias and variance predictions after increasing the model order.

- On the impact of selected order on results. You were talking about a related bias/variance trade off. So, do you know in the literature some solutions to avoid overfitting? Can you cite something? Can you use something in your case to test for a good order of system?

If we evaluate the step input estimation bias and variance obtained by the data-driven method for different model orders, we expect the results follow the bias-variance tradeoff. It is commonly accepted that for decreasing orders the bias is large since not all dynamics are captured, and that for increasing orders the bias is small and the variance is large because the system dynamics are captured and also the measurement noise is being modeled. However, after observing the results in the experimental validation of the method, we get better performance by overfitting. This could be valid only for some estimators. For example, in the research work of [17] it is described a neural network that does not follow the bias variance tradeoff, contradicting [18] and [19], who state that an overparameterized neural network reduces the bias by interpolating the data with a significant variance contribution to the mean-squared error.

Some approaches described in the literature to avoid overfitting include crossvalidation (CV), minimum description length (MDL) and the Akaike information criterion (AIC). These methodologies are described in books such as [20] and [21]. In the paper [22], CV and MDL are compared trying to answer whether penalized criteria in classical AIC should be preferred to CV for a given model selection problem. On one hand, provided the data is i.i.d., CV gives good model selection in almost any framework. On the other hand, CV can be less accurate, even with larger computational cost, than procedures designed to be optimal in a specific framework, such as AIC.

To describe the classical methods, the following paragraph has been added in subsection 5.2.

There exist methods for estimating the performance of different models that are useful to choose among the models. The model selection methods can be analytical, such as the Akaike information criterion (AIC) introduced in the paper of [23], the minimum description length (MDL) initiated by the work of [24], or numerical such as crossvalidation. The AIC is defined for a class of models fit by maximum likelihood. For the linear regression problem with Gaussian perturbation (3.20), the AIC is given by

$$\text{AIC} = 2(n + 1) - \ln(\hat{L}) \quad (5.9)$$

where  $n + 1$  is the number of estimated parameters, and  $\hat{L}$  is the maximum value of the likelihood function of the model. The AIC rewards the goodness of fit by considering the maximum likelihood value, and adds a penalty term that increases the cost with respect to the number of parameters to prevent overfitting. In the opinion of [25], the AIC is reliable when the priors overwhelmed by the likelihood, the posterior distribution is approximately multivariate Gaussian, and the sample size is much larger than the number of parameters. Using this criterion for model selection is simple, [20] suggest just picking the model that provides the smallest AIC value.

The MDL approach is motivated from an optimal coding viewpoint, aiming to choose the most parsimonious model, that, according to [20], is the shortest model that can encode the data for transmission of messages. If we have a model with parameters  $\theta$ , and data  $(K, y)$ ,

with the conditional probability of the outputs  $P(y|\theta, K)$ , and assuming the receiver knows the inputs and we transmit the outputs, then the message length required to transmit the outputs is

$$\text{length} = -\ln(P(\theta)) - \ln(P(y|\theta, K)). \quad (5.10)$$

The length is the sum of the code length required for transmitting the model parameters with the code length required for transmitting the discrepancy between the model and the actual target values. The recommended model is the one that minimizes (5.10). [25] argues that model fitting is a form of data compression, where parameters explain the relationships among the data. The parameters compress the data into a simpler form, with some loss of information about the sample, and can be used to generate new data by decompression. Learning about the data requires using a simpler model that achieves some compression, but not too much.

- Do you know of cross-validation: split data into two parts or several parts; some for estimation, some for validation? Have you tried this? Can this be used here to select good orders? Better results? Influence of system order on bias/variance; if I only use one dataset it is complicated and usually you can increase complexity to mimic a data set, but to avoid overfitting (too high order) use a test/validation data set. with this i should see what is the best value... Have you tried this?

**The crossvalidation method has not been applied on the experimental data. This data contains the step responses of a weighing system. The step responses are acquired with a sampling frequency of 4 kHz. One difficulty for applying cross-validation is the division of the data. If we select the first 2/3 of the step response samples for order estimation and the last 1/3 for validation, we may leave the data with the diminished transient effects in the validation set. Another approach could be taking every second sample for validation, dividing the measured responses in two by downsampling. However, this approach requires filtering against possible aliasing because the frequency characteristics of the response signal change. A third way that could be tried is dividing the 100 measured responses in two sets, devoting 80 responses for order estimation and 20 responses for validation.**

- In Section 6.2.2. Why do you call it a max likelihood approach? To me it is not one. Why do you call it like this? Where is the likelihood? Depending on which likelihood (eg. Gaussian, ...) it leads to LS. Don't call it ML if you don't give distributions on ... In this case it is better to call it a nonlinear minimization procedure. Not clear why you call it ML.

**The method description in Section 6.2.2 has been modified to explain why the method is a maximum likelihood method.**

- Concerning the ML approach in simulations, you say that it takes 30 seconds to complete. What is done exactly in these 30 seconds; how many samples, realizations,... I am a bit surprised you require 30 sec for 2 parameters.

**The large time consumption in the local optimization iterations is due to the different functions that take place during the minimization of the error function. The iterations are performed in a receding horizon scheme, starting from a small amount of samples (50) and repeating the error minimization for each new subset of samples acquired (5), until covering all samples (1000). In each iteration, the**

response of a dynamic system is simulated several times, for the different tried optimization variables. This simulated response is compared to the actual measured response, aiming to reduce the error between these two responses. Every time that a sensor response is simulated for a particular vector of optimization variables being tested, the analytic Jacobian is also evaluated. The evaluation of the Jacobian matrix requires the simulation of one additional dynamic system per to-be-optimized parameter. Considering that the simulations of all these responses take considerable amount of time, the elapsed time that the ML method takes is 25 s in average. Then, there is a need for efficient methods that could run in real time to optimize online the input model parameters and the sensor model parameters of interest.

## Nikos Deligiannis

- When solving algorithms. Do you opt for closed-form solutions, or do you use optimization based algorithms? How to solve LS problems? Do you solve them using pseudo-inverse or do you do gradient descent?

The closed form solutions allow us to describe the operations in a more elegant way. The closed form solutions implies batch processing of the operations. The formulation of the problem also gives a hint of which solution to choose. For instance, an unconstrained linear minimization problem can be solved with least squares but constrained nonlinear minimization problems most likely require optimization. There are different methods to solve least squares problems. The pseudo-inverse approach is the first option that comes to my mind, but modern numerical software, such as Matlab and julia, have toolboxes that solve efficiently the least squares problems depending on the regression matrix and regressor data. Regarding the maximum-likelihood method of Chapter 6, the optimization method uses gradient descend to find the direction towards the local minimum. Other minimization problems in low-rank approximation can be solved using variable-projection optimization methods.

- Today there is an overwhelming amount of data sensed and communicated; How would you apply your method in a scenario where dimensionality increases significantly? This relates to the questions of previous jury members; what is the implication of using numerical versus closed form solutions?

The increase in dimensionality is explained by the current trend in the industry of installing more sensors. More signals are used to monitor the production lines. The transference of data is relevant for decision making. The data-driven input estimation method can be used to estimate recursively the input applied to the sensors, and to tract the evolution of inputs. In this way, the estimation method facilitates the controllers by providing online estimations of the inputs. This improves the performance of closed form solutions, which need more computational resources and time more time for the batch processing of data.

- What if the  $K^T K$  matrix is not invertible? What if you have fewer samples than unknowns; what can you do? Solutions that assume structure like in compressed sensing, regularization,... can you use them, what would be the impact of that?

A possible lack of invertibility of the matrix  $K^\top K$  may be due to missing data. A different methodology to low-rank approximation is compressed sensing, where the missing data is interpreted as sparsity. Under the assumption that a signal is sparse because it is subject to many near zero coefficients in a given basis, then changing to another basis and taking a weighted linear combination of the available samples would lead to the estimation of the missing data values. This idea is described as future work in the final chapter of the thesis

## Compressed sensing methods for recovering missing data

Compressed sensing is a methodology that is being developed quite fast and that can offer an alternative to perform measurements with missing data. The aim of compressed sensing is the recovery of the missing samples of a signal by computing a sparse coefficient vector, under the assumption that the signal belongs to a generating system of larger order. The missing data estimation problem can be expressed as an overdetermined system of equations with a sparse regression matrix. The review presented in [26] provides an introduction of the mathematical modeling of compressed sensing, and gives the conditions for sparse recovery. The recovery algorithms include convex optimization, greedy and combinatorial algorithms, to name a few.

According to [27], compressed sensing is applied when we cannot afford buying enough sensors like in non-visible wavelengths applications, when the measurements themselves are expensive as in neutron-scattering imaging, when measurements take a lot of time as in medical imaging, and when a constraint of the measurement is the power consumption. A particular field of application for compressed sensing is the recovery of bioelectrical signals, as it is detailed in [28].

Another good review of compressed sensing is found in [29], where it is explained that the research community is splitted into a group that studies the problems in which the compressed sensing matrices exhibit structure due to the measurement system modeling, and another group which considers that the signal structured representations not only depend on sparsity but instead assume that the signals belong to continuous time and have infinite-dimensional representations. The solutions provided exploit these structures in the development of the algorithms. Some examples of these structure matrices are described in [30] including Gaussian, Toeplitz, Hadamard, and structurally random matrices.

It might be interesting to compare the solutions of the low-rank approximation methods with compressed sensing in terms of computational complexity and feasibility for online implementations. Therefore, it is proposed as a future research to develop a compressed sensing algorithm, that can estimate directly the input true value by processing the sensor response to the unknown input, and in presence of missing data.

- You mention that you improve complexity with your approach, but I missed a complexity analysis. Did you do this? Anything in processing time, flops,...?

A discussion on the scalability of the data-driven step input estimation method has been added in Section 3.2. The method uses recursive least squares to update the input estimation value, thus its complexity is linear in the number of parameters  $n + 1$ , and independent from the number of samples  $N$ .

The computation of the bias and variance of the input estimation, which is described in Chapter 4, has a complexity  $O(n^3 N)$ . This complexity prevents the

online implementation of the estimation bias and variance. It is proposed as future research work to study efficient methods to estimate these statistical moments. The objective of this future work is to estimate the input and its mean squared error in real time.

- You mentioned deep learning could be applied; could you show how? An issue is that neural nets are trained in high SNR data, but applied in engineering with lower SNR. This clogs performance; any ideas? Is this a limitation?

Deep learning could be used but it would be an overkill methodology for linear estimation. However, for metrology applications where the input is the to-be-estimated signal, implementing deep learning could be a challenge. The neural networks are suitable for non-linear model based applications where the training requires the design and application of input stimuli to correlate with the corresponding outputs of the system. In metrology, it is not possible to train deep learning methodologies with real input data, because the input data is unknown and cannot be observed. By applying neural networks trained by simulation would estimate inaccurately the desired inputs.

- You mention that you consider Kalman filters; have you seen approaches where they write Deep Learning in form of Kalman Filter; Could this be applicable?

The Kalman filter is mentioned in Chapter 3 to give context for the data-driven step input estimation method. The estimation of the input from its corresponding sensor step response in presence of noise can be estimated with a Kalman filter when we have a model of the sensor.

Regarding deep learning and Kalman filters, [31] provides a chronological review of shallow and deep learning methods considering the depth of the used chains of possible learnable, causal links between actions and effects. Deep learning is about assigning credit across the computational stages of neural networks. Deep Kalman filters are filters that incorporate neural networks to improve the performance when the system under study is nonlinear, or to mitigate the model uncertainties. To mention a few deep Kalman filters, we have [32] that fuses global positioning sensors and inertial sensors responses to model non-linear, time-variant, and highly correlated inertial sensors errors, [33] fuses camera images with inertial sensor measurements to estimate the camera pose, [34] designs a Kalman filter from a bottom-up approach to forecast the short-term load of a house considering the large uncertainty in the human interaction and weather conditions, [35] uses a neuron-based nonlinear autoregressive model to adapt the Kalman filter variables for a robot positions tracking, [36] reconstructs the particle paths in the large hadron collider with a recurrent neural network that extrapolates, builds, and evaluates path candidates, [37] combines large state spaces with nonlinear Q-function approximations to compensate for the uncertainty in the neural network weight values, and [38] proposes an algorithm for learning causal generative temporal models from high-dimensional noisy data, such as the evolution of a large number of patients suffering from a particular illness, to enhance Kalman filters under arbitrary complex transition dynamics.

The problem under study in this thesis is the linear estimation of a step input with unknown level given the sensor step response. The data-driven input estimation

method is recursive and assumes that the model of the LTI sensor is not available. The deep Kalman filters are suitable for non-linear applications where a model of the input is available, and the incorporated neural networks could estimate also the sensor model parameters. To do this, the computational complexity of deep Kalman filters may be larger than that of the data-driven input estimation method.

- Recommendation: if you revise the thesis: maybe change "I propose" to "we propose". The formulation "I propose" was replaced by "we propose".

## Philippe Dreesen

- Your main assumption is that a sensor outputs a steady state value that is proportional to the to-be-measured quantity. This means that you assume that the sensor is a linear system. What if this assumption is not met, and the sensor has a saturation for high values? Can you still use your methods?

If the sensor has a saturation for high input values, the data-driven step input estimation method still cannot provide an estimation of the input step level. To perform the estimation with missing data, it is necessary to complete the Hankel matrix in the estimation problem formulation. If the sensor response were exact data, it would be enough to rearrange the available data from the Hankel matrix and find the missing data. Nevertheless, this procedure would take considerable time. Moreover, under noise perturbation, efficient approximation algorithms are needed to perform the matrix completion in real time. The last paragraph in Chapter 3 locates the data-driven step input estimation method in perspective to the most powerful unfalsified model of the behavioral approach theory.

**Note.** The formulation of the data-driven step input estimation method is based on the behavioral theory introduced in [39]. A behaviour of a dynamical system is any set of input-output trajectories that belong to the system. It is demonstrated in [40] that when an input-output trajectory is persistently exciting enough, a Hankel matrix constructed from this trajectory represents the most powerful unfalsified model. This model is unfalsified because it describes exactly all possible trajectories of the system, and is the most powerful because it is the system representation with the minimum complexity. The rank of the Hankel matrix constructed from a trajectory is a metric used to find the order of persistency of the trajectory. The rank-deficient Hankel matrix with the smaller number of block rows indicates the trajectory order of persistency. When the system is excited by a step input, the system is equivalent to an autonomous augmented system, and a model for it can be obtained from the Hankel matrix constructed only from the step response.

Also, in the future work section is described the need for efficient optimization algorithms that could solve the low-rank approximation problem allowing for an online estimation of the step input value.

### The data-driven input estimation methods with missing data

The implementation of the data-driven step input estimation method can be compromised when one or more step response samples are not properly acquired. The missing data in



the construction of the Hankel matrix may be due to sample transference errors, or due to sensor saturation problems. The saturation values in the sensor response are not valid for the computation of the step input level because they violate the linearity assumption of the sensor. The solution to the most powerful unfalsified model identification problem in presence of missing data was provided in [41]. The methods presented in this work are formulated as a matrix completion with the constraint that the to-be-completed matrix must be a rank deficient Hankel matrix. The low-rank approximation of Hankel structured matrices with missing data has been studied in [42], where the proposed solution is based on a variable-projection method that requires optimization. The software for this method, including a link to obtain it, is described in [43]. This method is applicable in matrix completion and system identification problems and has inspired the development of the signal recovery method in for nuclear magnetic resonance spectroscopy described in [44], and data compression and feature recognition in [45] based on sparse low-rank matrix approximation.

The solutions of the low-rank approximation with missing data problems are found by iterative optimization and, therefore, require large amount of computational power. In order to implement data-driven input estimation methods with missing data, there is a need for efficient methods of small complexity  $O(n)$  that solve online the optimization problem. This is an open problem for future research.

- Sensors are physical devices and deliver a continuous-time output. The models that you use are in discrete-time. Are there any risks that you miss information by not sampling fast enough?

**In Chapter 3 has been added a discussion on the step invariance transformation. This defines the conditions of the input for describing exactly the system dynamics in discrete time after sampling the continuous system response. If the input is stepwise continuous, then the zero order hold sampling of the step response leads to an exact discrete time representation of the system. The method under study in this thesis was developed for step input, then we can rely on the discrete time representation. There exist corresponding invariant transformations for piecewise linear inputs and for piecewise polynomial inputs that lead to exact representations from first and second order sampling of the system response, respectively.**

- Related to earlier questions about compressed sensing. There are some recent results on the connections between sparsity and low-rank Hankel matrix completion and approximation. Say that you have an issue with missing data, e.g., from sensor outages. Would it be possible to plug in a procedure that does structured matrix completion and approximation into your methods?

**Unfortunately, there is still a need for efficient optimization algorithms that could solve the low-rank approximation problem allowing for a online estimation of the step input value.**

## Roger Vounckx

- Two practical questions. On page 35-49 you consider a weight of 138.x grams. You sample relatively slowly (4 kHz) and after 500 samples we can consider that we have correct esti-

mation. If I have a container that fills up and weight is constantly changing. Can we tackle this?

**Yes, that is exactly the idea considered by the affine input estimation method. Since the data-driven estimation method can track the changes of the input, the parameters of an input model that describes the constant variation can be recovered if we add the forgetting factor to the recursive solution algorithm.**

- Consider an optical fiber link with a detector capable of 100 million measurements per second. Can I speed up the link by using your method? You have to do calculations: how high can you go in speed before running into troubles with DSP?

**The optical fiber link transmits digital pulses. Since the period would be 1 ns, the step input method implementation would require to sample in much shorter time. Assuming that the duty cycle is 50 %, and that the number of samples required for the step detection are 200, then the sampling time should be 1.25 fs or lower. This could be a constraint for the commutation circuits and we would be reaching a feasibility limit. However, theoretically the method would perform detecting the pulses level changes.**

## Rik Pintelon

- Returning to a question by Guillaume Mercere: on page 17; do you need normality? You made 2nd order Taylor series expansion. Which moment do you need to estimate? Look at the equations page 16. There are 3rd, 4th order moments,... you need to know something more about the noise; this is one of the reasons to make Gaussian assumption: 3rd order moment is 0 (symmetry); 4th order for Gaussian is easy to have an explicit relationship.

**Section 4.1 has been appended with a paragraph to discuss on the need for the normality assumption.**

The normality assumption of the perturbation noise is necessary to provide more prior information into the method formulation. The second order Taylor series expansion (??) is developed considering the perturbation noise that gets into the regression matrix from the sensor transient response. Assuming only that the perturbation noise is distributed with zero mean and variance  $\sigma_\epsilon^2$  is not enough. If we additionally assume normality for the perturbation noise, then we benefit from the knowledge of the third moment equal to zero due to symmetry, and the fourth moment equal to three times the squared variance.

- Returning to a question by Philippe Dreesen on missing data. If you can deal with missing data, you can deal with saturation in sensors: you just remove the saturations. What is the rank of the Hankel matrix? Is it full rank or not? The question is if matrix completion can help? It works in low-rank approximations; not if the matrix is of full rank.

**The last paragraph in Chapter 3 was added to describe the position the data-driven step input estimation method with respect to the most powerful unfalsified model of the behavioral approach theory. In the future work section, it was added a description of the need for efficient optimization algorithms that could solve the low-rank approximation problem enabling the online estimation of the step input value.**

## Ivan Markovsky

- About scalability of algorithms: You said in the very beginning that it can be very fast compared with system identification. This is the point: can you think of a good answer for these questions on scalability? Nikos Deligiannis mentioned computational analysis. You can directly say for (R)LS what is the computational complexity. Order three if full matrix inversion, now per iteration only linear. So you can scale to very large data sizes.

**One paragraph at the end of Section 3.2 has been included -to discuss the scalability of the data-driven step input estimation method.**

The computational complexity of the RLS algorithm solution to the system of equations (3.15) is  $O\left((n+1)^2\right)$ . The largest computational requirement is for the initialization of the algorithm, where a matrix inverse is required when the number of samples is just enough to have a square matrix  $\tilde{\mathbf{K}}_{n+1}$ . From there on, the RLS algorithm updates the solution with linear complexity. Therefore, even though a large number of samples  $N$  makes the matrix in the system of equations (3.15) increase in the number of rows, the estimation of the solution does not increase in complexity. The data-driven step input estimation method is then scalable for any sensor of order  $n$ , and can be executed in devices with limited computational resources, provided that the computation of the  $(n+1) \times (n+1)$  inverse matrix is feasible.

## Detailed comments provided by jury members

### Guillaume Mercere

- It is important for me to know why these new developments are generated. That is the reason why I suggest adding sections or paragraphs dedicated to
  - the reasons why, in the industry, we need such a new technique. As I said during the defense, do you have a list of sensors for which waiting for the steady regime for getting the measurement is not possible? Why do you need a fast algorithm? What are the reasons why industry people need fast algorithms? In the end, it seems that, for you, it is more important to have a fast biased result than an unbiased estimate. Why? What do you mean by fast?

The Introduction has been appended with illustrative examples in which the measurements would benefit from the direct estimation of the input. The sensors in this list are those with a low bandwidth due to its construction and principle of operation. The transient stage of these sensors is long, from a practical point of view, because it takes relatively long time for the transient effects to decrease below the noise floor. From that moment on, the input value can be estimated easily since it is proportional to the sensor response in steady regime. The estimation of the input is classically obtained through filtering the sensor response with a dynamic system designed to perform a deconvolution. The response of the additional system approximates the input value, but unfortunately there is also a transient response with a duration that is expected smaller than that of the actual sensor. Then, there is a trade-off between the estimation accuracy and the estimation time. The compensation time can be still improved further with signal processing

methods that are independent of sensor models. These data-driven methods can be conceived as adaptive methods with shorter transient time than classical compensation systems. The data-driven step input estimation method formulates the estimation problem as an errors-in-variables problem, and proposes a solution with recursive least squares. This implies that we have a faster response with an asymptotic bias. Future research can be devoted to find a fast and unbiased online solution to the errors in variables system.

- the literature on EIV estimation problems. As shortly mentioned in Chapter 2, there are solutions in the literature (by the way, you never mention the recent book written by Torsten Soderstrom in your document which is strange for me). But you do not describe them (even shortly) and, more importantly, you do not criticize them. Not in general but, at least, explain why you do not use them, what are their advantages and drawbacks and, finally, w.r.t. this list of positive and negative points, why you choose a (R)LS solution. E.g., why not using a recursive instrumental variable solution instead?

A description of IV methods has been added in Section 4.1.4

#### 4.1.4. Instrumental variables formulation of the data-driven step input estimation method

The errors-in-variables problem formulation (3.21) of the data-driven step input estimation method can be converted into an instrumental-variables problem. A guide to do this conversion is found in [9] and in [46]. The instrumental variable  $\tilde{\mathbf{Z}}$  has to be selected so that the normal equations (3.22) are expressed instead as

$$\tilde{\mathbf{Z}}^\top \tilde{\mathbf{K}} \boldsymbol{\theta} = \tilde{\mathbf{Z}}^\top \tilde{\mathbf{y}}. \quad (4.23)$$

The selection of the instrumental variable has to be done in such a way that is correlated with the regression matrix  $\tilde{\mathbf{K}}$ , and uncorrelated with the perturbations  $\boldsymbol{\epsilon}$ . Assuming that the block-Hankel matrix  $\tilde{\mathbf{K}}$  is given by

$$\tilde{\mathbf{K}} = \begin{bmatrix} G & \Delta \tilde{y}(1) & \Delta \tilde{y}(2) & \cdots & \Delta \tilde{y}(n) \\ G & \Delta \tilde{y}(2) & \Delta \tilde{y}(3) & \cdots & \Delta \tilde{y}(n+1) \\ \vdots & \vdots & \vdots & & \vdots \\ G & \Delta \tilde{y}(n+1) & \Delta \tilde{y}(n+2) & \cdots & \Delta \tilde{y}(2n) \end{bmatrix}, \quad (4.24)$$

then, the instrumental variable  $\tilde{\mathbf{Z}}$  should not contain the elements that construct  $\tilde{\mathbf{K}}$ , i.e., the elements of the vector  $(\tilde{y}(1), \dots, \tilde{y}(2n+1))^\top$ . A possibility to generalize this idea is expressing the instrumental variable as

$$\tilde{\mathbf{Z}}(t) = \begin{bmatrix} G & \Delta \tilde{y}(t-4n) & \Delta \tilde{y}(t-4n+1) & \cdots & \Delta \tilde{y}(t-3n-1) \\ G & \Delta \tilde{y}(t-4n+1) & \Delta \tilde{y}(t-4n+2) & \cdots & \Delta \tilde{y}(t-3n) \\ \vdots & \vdots & \vdots & & \vdots \\ G & \Delta \tilde{y}(t-3n) & \Delta \tilde{y}(t-3n+1) & \cdots & \Delta \tilde{y}(t-2n-1) \end{bmatrix}, \quad (4.25)$$

for a regression matrix

$$\tilde{\mathbf{K}}(t) = \begin{bmatrix} G & \Delta \tilde{y}(t-2n+1) & \Delta \tilde{y}(t-2n+2) & \cdots & \Delta \tilde{y}(t-n) \\ G & \Delta \tilde{y}(t-2n+2) & \Delta \tilde{y}(t-2n+3) & \cdots & \Delta \tilde{y}(t-n+1) \\ \vdots & \vdots & \vdots & & \vdots \\ G & \Delta \tilde{y}(t-n+1) & \Delta \tilde{y}(t-n+2) & \cdots & \Delta \tilde{y}(t) \end{bmatrix}, \quad (4.26)$$

for  $t = 4n + 1, \dots, N$ .

The difference with the EIV formulation is that this implementation of the IV method requires to wait more, until  $4n + 1$  samples of the sensor response are read, instead of  $2n + 1$  samples. For sensors of small order this twice-the-time waiting does not represent a big change, but as the sensor order increases, the need for waiting can turn the method into impractical since a lot of time should pass before starting to estimate the input.

There exists a generalized instrumental variables method (GIVE) that was proposed to identify dynamic models from input/output data. GIVE is a bias-eliminating method that uses a generalized instrumental vector constructed from delayed values of the input and output signals [9]. Since the aim of this thesis is to analyze a method that directly estimates the input from the observed sensor output, without identifying explicitly a dynamic model of the sensor, the GIVE method is out of the scope of this work. A future research can include studying the formulation of a data-driven input estimation method in the spirit of the GIVE method. Nevertheless, a brief description of the GIVE method and its recursive version is given in the following paragraphs.

#### 4.1.4.1 Generalized instrumental variables method

For an errors-in-variables problem expressed as

$$\tilde{y}(t) = \tilde{\phi}(t)\theta \quad (4.27)$$

where  $\tilde{y}(t)$  is the output perturbed by measurement noise,  $\theta$  is the vector of to-be-estimated parameters, and  $\tilde{\phi}(t)$  is the regressor vector constructed from delayed samples of the measured input  $\tilde{u}(t)$  and output  $\tilde{y}(t)$ . The least-squares estimate  $\hat{\theta}_{\text{LS}}$  is the solution of the normal equation

$$\hat{\mathbf{R}}_{\tilde{\phi}} \hat{\theta}_{\text{LS}} = \hat{\mathbf{r}}_{\tilde{\phi}\tilde{y}}, \quad (4.28)$$

where  $\hat{\mathbf{R}}_{\tilde{\phi}}$  is the covariance matrix of the regressor vector, and  $\hat{\mathbf{r}}_{\tilde{\phi}\tilde{y}}$  is the cross-covariance matrix of the regressor vector and the output. When the number of samples becomes infinitely large, these covariances matrices can be expressed as a sum of the contributions from exact data and measurement noise contributions

$$\begin{aligned} \lim_{N \rightarrow \infty} \hat{\mathbf{R}}_{\tilde{\phi}} &= \mathbb{E}\{\tilde{\phi}\tilde{\phi}^\top\} = \mathbf{R}_{\phi} + \mathbf{R}_{\eta} \\ \lim_{N \rightarrow \infty} \hat{\mathbf{r}}_{\tilde{\phi}\tilde{y}} &= \mathbb{E}\{\tilde{\phi}\tilde{y}^\top\} = \mathbf{r}_{\phi y} + \mathbf{r}_{\eta\epsilon}, \end{aligned} \quad (4.29)$$

where we are assuming that  $\tilde{y} = y + \epsilon$ ,  $\tilde{\phi} = \phi + \eta$ , and  $\mathbf{r}_{\phi y} = \mathbf{R}_{\phi}\theta$ . The bias of the least-square solution can be expressed as

$$\mathbf{R}_{\tilde{\phi}} \left( \hat{\theta}_{\text{LS}} - \theta \right) = \mathbf{r}_{\tilde{\phi}\tilde{y}} - \mathbf{R}_{\tilde{\phi}}\theta = \mathbf{r}_{\eta\epsilon} - \mathbf{R}_{\eta}\theta, \quad (4.30)$$

If we assume that the measurement noise is white for both the input and output,

$$\mathbf{r}_{\eta\epsilon} = \mathbb{E}\{\eta\epsilon^\top\} = \mathbf{0} \quad \text{and} \quad \mathbf{R}_{\eta} = \begin{bmatrix} \sigma_{\epsilon}^2 \mathbf{I}_{n_y} & \mathbf{0} \\ \mathbf{0} & \sigma_u^2 \mathbf{I}_{n_u} \end{bmatrix}, \quad (4.31)$$

where  $\sigma_{\epsilon}^2$  and  $\sigma_u^2$  are the variances of the output and the input, respectively. The normal equation (4.28) implies that

$$(\mathbf{R}_{\phi} + \mathbf{R}_{\eta}) \left( \hat{\theta}_{\text{LS}} - \theta \right) \neq \mathbf{0}, \quad (4.32)$$

and the estimate  $\hat{\boldsymbol{\theta}}_{\text{LS}}$  is not consistent. Instead, a bias-compensating least squares estimation can be obtained with

$$\hat{\boldsymbol{\theta}}_{\text{BCLS}} = \left( \mathbf{R}_{\tilde{\boldsymbol{\phi}}} - \mathbf{R}_{\boldsymbol{\eta}} \right)^{-1} \left( \mathbf{r}_{\tilde{\boldsymbol{\phi}}\tilde{\mathbf{y}}} - \mathbf{r}_{\boldsymbol{\eta}\epsilon} \right). \quad (4.33)$$

To formulate the generalized instrumental variable estimator, a generalized instrumental vector  $\mathbf{z}(t)$  is built from delayed samples of the input and the output. The dimension of the generalized instrumental vector should be  $n_{\mathbf{z}} \geq \dim(\boldsymbol{\vartheta}) = n_u + n_y + 2$ , since the total parameter vector  $\boldsymbol{\vartheta} = (\boldsymbol{\theta}^\top, \boldsymbol{\rho}^\top)^\top$  considers also the variances of the measurement noise  $\boldsymbol{\rho} = (\sigma_\epsilon^2, \sigma_u^2)^\top$ . Since the generalized instrumental vector  $\mathbf{z}(t)$  and the equation error  $\tilde{y}(t) - \tilde{\boldsymbol{\phi}}(t)\boldsymbol{\theta}$  must be correlated, we can express the system of equation as

$$\left( \mathbf{R}_{\tilde{\mathbf{z}}\tilde{\boldsymbol{\phi}}} - \mathbf{R}_{\zeta\boldsymbol{\eta}}(\boldsymbol{\rho}) \right) \boldsymbol{\theta} = \left( \mathbf{r}_{\tilde{\mathbf{z}}\tilde{\mathbf{y}}} - \mathbf{r}_{\zeta\epsilon}(\boldsymbol{\rho}) \right), \quad (4.34)$$

where  $\tilde{\mathbf{z}} = \mathbf{z} + \boldsymbol{\zeta}$ .

#### 4.1.4.2 Recursive generalized instrumental variables method

If we define

$$\begin{aligned} \mathbf{h}(\boldsymbol{\rho}, \boldsymbol{\theta}) &= \mathbf{r}_{\zeta\epsilon}(\boldsymbol{\rho}) - \mathbf{R}_{\zeta\boldsymbol{\eta}}(\boldsymbol{\rho})\boldsymbol{\theta} = \mathbf{J}(\boldsymbol{\theta})\boldsymbol{\rho}, \\ \mathbf{R}(t) &= \frac{1}{t} \sum_{\tau=1}^t \tilde{\mathbf{z}}(\tau) \tilde{\boldsymbol{\phi}}^\top(\tau), \\ \mathbf{q}(t) &= \frac{1}{t} \sum_{\tau=1}^t \tilde{\mathbf{z}}(\tau) \tilde{y}(\tau) \\ \mathbf{b}(t) &= \mathbf{q}(t) - \mathbf{h}(\boldsymbol{\rho}, \boldsymbol{\theta}), \end{aligned} \quad (4.35)$$

then the equation (4.34) is equivalent to

$$\mathbf{R}(t)\boldsymbol{\theta} = \mathbf{b}(t). \quad (4.36)$$

This representation allows a recursive form

$$\begin{aligned} \mathbf{R}(t+1) &= \mathbf{R}(t) + \frac{1}{t+1} \left( \tilde{\mathbf{z}}(t+1) \tilde{\boldsymbol{\phi}}^\top(t+1) - \mathbf{R}(t) \right), \\ \mathbf{b}(t+1) &= \mathbf{b}(t) + \frac{1}{t+1} \left( \tilde{\mathbf{z}}(t+1) \tilde{y}(t+1) - \mathbf{b}(t) - \mathbf{h}(\boldsymbol{\rho}, \boldsymbol{\theta}) \right), \\ \hat{\boldsymbol{\theta}}(t+1) &= \left( \mathbf{R}^\top(t+1) \mathbf{R}(t+1) \right)^{-1} \mathbf{R}^\top(t+1) \mathbf{b}(t+1) \\ \mathbf{q}(t+1) &= \mathbf{q}(t) + \frac{1}{t+1} \left( \tilde{\mathbf{z}}(t+1) \tilde{y}(t+1) - \mathbf{q}(t) \right) \\ \hat{\boldsymbol{\rho}}(t+1) &= \mathbf{J}^\dagger \left( \hat{\boldsymbol{\theta}}(t) \right) \left( \mathbf{q}(t) - \mathbf{R}(t) \hat{\boldsymbol{\theta}}(t) \right), \end{aligned} \quad (4.37)$$

where  $\mathbf{J}^\dagger \left( \hat{\boldsymbol{\theta}}(t) \right)$  is the pseudo-inverse of  $\mathbf{J}(\boldsymbol{\theta})$ .

The complexity of this recursive generalized instrumental variables estimator is  $O(n^3)$ , that is larger than the  $O(n^2)$  of the recursive least squares. Another drawback of the recursive generalized IV estimator is that it computes a pseudo-inverse matrix in each

iteration step. If we would need to formulate the data-driven step input estimation problem in the generalized IV framework, it would require to study the formulation in detail because currently the IV framework is based on the assumption that both measurements of the input and output are available, but for the step input estimation problem only the output is available. Therefore, it is not straightforward to implement the recursive generalized IV estimator to estimate the step input, and this can lead to future research.

- In the document, you always talk about recursive implementations of least squares solution but, in the end, you never give access to any recursive algorithm description (as you do for what you call the ML method). Why? Do it as well for the recursive weighted least squares solution.

**The following recursive least-squares algorithm description has been added to Section 3.2:**

The problem (3.20) admits a least-squares (LS) solution with normal equations given by

$$\tilde{\mathbf{K}}^\top \tilde{\mathbf{K}} \boldsymbol{\theta} = \tilde{\mathbf{K}}^\top \tilde{\mathbf{y}}. \quad (3.22)$$

The closed form of the LS solution can be expressed as

$$\hat{\boldsymbol{\theta}} = \tilde{\mathbf{K}}^\dagger \tilde{\mathbf{y}} = (\tilde{\mathbf{K}}^\top \tilde{\mathbf{K}})^{-1} \tilde{\mathbf{K}}^\top \tilde{\mathbf{y}}, \quad (3.23)$$

where  $\tilde{\mathbf{K}}^\dagger$  is the pseudo-inverse matrix of  $\tilde{\mathbf{K}}$ . As the number of samples  $N$  increases, the number of rows of the matrix  $\tilde{\mathbf{K}}$  grows, making the pseudo-inverse computation inefficient since it requires larger flops, memory and time.

For metrology applications, it is desired to have a fast solution that can be obtained in real-time. The recursive algorithm (RLS) is a least-squares alternative to implement the step input estimation method in real-time. The RLS recursively updates the solution of the system of equations, for each newly acquired sensor response sample, considering the previous value of the estimation, instead of performing a matrix inversion. An expression of the RLS equations is given in [47] as

$$\begin{aligned} \hat{\boldsymbol{\theta}}(t) &= \hat{\boldsymbol{\theta}}(t-1) + \boldsymbol{\kappa}_t \left( \tilde{y}(t) - \tilde{k}_t \hat{\boldsymbol{\theta}}(t-1) \right), \\ \boldsymbol{\kappa}_t &= \boldsymbol{\Psi}(t-1) \tilde{k}_t^\top / \left( 1 + \tilde{k}_t \boldsymbol{\Psi}(t-1) \tilde{k}_t^\top \right) \\ \boldsymbol{\Psi}(t) &= \left( \mathbf{I} - \boldsymbol{\kappa}_t \tilde{k}_t \right) \boldsymbol{\Psi}(t-1), \end{aligned} \quad (3.24)$$

where  $\tilde{k}_t$  represents the row of  $\tilde{\mathbf{K}}$  that corresponds to the  $t$ -th sample,  $\boldsymbol{\kappa}_t$  is a gain scalar,  $\boldsymbol{\Psi}(t)$  is a covariance matrix, and the estimates and the covariance matrix are initialized respectively with  $\hat{\boldsymbol{\theta}}(1) = \hat{\boldsymbol{\theta}}(0)$  and  $\boldsymbol{\Psi}(-1) = \boldsymbol{\Psi}(0)$ . Since the RLS is a recursive implementation of the least-squares computation, the RLS solution is equivalent to the LS solution. Moreover, the statistical analysis of the LS solution is valid also for the RLS solution, and is preferred since it is simpler to express the statistics using the closed form of the LS solution.

**The following description for the exponentially recursive least-squares algorithm has been added to Section 6.2.1:**

The exponentially weighted recursive least-squares (RLS) can solve recursively the estimation problem (3.21). In [47] is described the exponentially weighted RLS algorithm, which is as

follows

$$\begin{aligned}\hat{\boldsymbol{\theta}}(k) &= \hat{\boldsymbol{\theta}}(k-1) + \boldsymbol{\kappa}_k \left( \tilde{y}(k) - \tilde{\mathbf{k}}_k^T \hat{\boldsymbol{\theta}}(k-1) \right), \\ \boldsymbol{\kappa}_k &= \omega^{-1} \boldsymbol{\Psi}(k-1) \tilde{\mathbf{k}}_k^T / \left( 1 + \omega^{-1} \tilde{\mathbf{k}}_k^T \boldsymbol{\Psi}(k-1) \tilde{\mathbf{k}}_k \right) \\ \boldsymbol{\Psi}(k) &= \omega^{-1} \left( \mathbf{I} - \boldsymbol{\kappa}_k \tilde{\mathbf{k}}_k^T \right) \boldsymbol{\Psi}(k-1),\end{aligned}\tag{6.4}$$

for  $k = 2n+1, 2n+2, \dots$ , where  $\tilde{\mathbf{k}}_k$  represents the row of  $\tilde{\mathbf{K}}$  that corresponds to the  $k$ -th sample,  $\boldsymbol{\kappa}_k$  is a gain scalar, and  $\boldsymbol{\Psi}(k)$  is a covariance matrix. The estimates and the covariance matrix are initialized using the first  $n+1$  samples, i.e.,

$$\begin{aligned}\hat{\boldsymbol{\theta}}(2n+1) &= \left( \tilde{\mathbf{K}}_{n+1}^T \boldsymbol{\Omega} \tilde{\mathbf{K}}_{n+1} \right)^{-1} \tilde{\mathbf{K}}_{n+1}^T \boldsymbol{\Omega} \tilde{\mathbf{y}}_{n+1}, \quad \text{and} \\ \boldsymbol{\Psi}(2n+1) &= \left( \tilde{\mathbf{K}}_{n+1}^T \boldsymbol{\Omega} \tilde{\mathbf{K}}_{n+1} \right)^{-1},\end{aligned}\tag{6.5}$$

where  $\tilde{\mathbf{K}}_{n+1}$  is the matrix  $\tilde{\mathbf{K}}$  with the first  $n+1$  rows, and  $\tilde{\mathbf{y}}_{n+1}$  is the vector  $\tilde{\mathbf{y}}$  with the first  $n+1$  elements. When  $\omega = 1$ , the exponentially weighted RLS solution is identical to that of the RLS. When  $\omega < 1$ , the older residuals are weighted with lower values than the residuals of recent observations. In this way, the solution of the minimization problem depends more on newer data, and less on older data.

- Double check your equations page 12 and 13. By the way, can you explain where Eq. (3.6) comes from?

**The equations in page 12 and 13 have been reviewed and typos corrected. The derivation of Eq. (3.6) has been improved with more explanation:**

A third method can directly estimate the input step level from the step response, without identifying a sensor model. To derive this method, we use the first difference operator,  $\Delta = \sigma - 1$ , where  $\sigma$  is the shift operator, defined as  $(\sigma^\tau y)(k) = y(k + \tau)$ . Applying the first difference operator to the system state-space representation (3.1) results in the autonomous system

$$\Delta \mathbf{x}(k+1) = \mathbf{A} \Delta \mathbf{x}(k), \quad \Delta y(k) = \mathbf{C} \Delta \mathbf{x}(k), \quad \text{with} \quad \Delta \mathbf{x}_{\text{ini}} = \Delta \mathbf{x}(0),\tag{3.11}$$

where  $\Delta u(k) = 0$ , for  $k \geq 0$ , and  $\Delta \mathbf{x}(0) = (\mathbf{A} - \mathbf{I})\mathbf{x}(0) + \mathbf{B}u$ .

If the response  $\Delta y$  of this autonomous system is persistently exciting of order  $L$ , then the rank of the Hankel matrix  $\mathcal{H}_{L+1}(\Delta y)$  of  $L+1$  block rows constructed from  $\Delta y$  satisfies

$$\text{rank}(\mathcal{H}_{L+1}(\Delta y)) = \text{rank} \left( \begin{bmatrix} \Delta y(1) & \Delta y(2) & \cdots & \Delta y(n) \\ \Delta y(2) & \Delta y(3) & \cdots & \Delta y(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ \Delta y(L+1) & \Delta y(L+2) & \cdots & \Delta y(L+n) \end{bmatrix} \right) \leq L,\tag{3.12}$$

and the Hankel matrix  $\mathcal{H}_{L+1}(\Delta y)$  is a linear map to the natural responses of the augmented autonomous system (3.4)

$$\mathbf{y}_{\text{natural}} = \mathcal{H}(\Delta y) \boldsymbol{\ell},\tag{3.13}$$

from any vector  $\boldsymbol{\ell} \in \mathbb{R}^n$ . The total response of the system is the sum of the forced response, due to the applied input  $u$  and the natural response

$$\mathbf{y} = \mathbf{y}_{\text{forced}} + \mathbf{y}_{\text{natural}} = G u + \mathcal{H}(\Delta y) \boldsymbol{\ell},\tag{3.14}$$



that is equivalent to

$$\underbrace{\begin{bmatrix} y(n+1) \\ \vdots \\ y(N) \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} G & \Delta y(1) & \Delta y(2) & \cdots & \Delta y(n) \\ G & \Delta y(2) & \Delta y(3) & \cdots & \Delta y(n+1) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ G & \Delta y(N-n) & \Delta y(N-n+1) & \cdots & \Delta y(N-1) \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} u \\ \ell \end{bmatrix}}_{\boldsymbol{\theta}}, \quad (3.15)$$

where  $N$  is the number of samples. From this point of view, the vector  $\ell$  is a linear transformation of the system's initial conditions.

- In Section 4.1.1 and 4.1.2, discuss the normality assumption. Please restructure Section 4.1.3 which is not clear for me.

**The discussion of the normality assumption has been added to Section 4.1, since it is relevant for the two cases considered. Section 4.1.3 has been restructured to improve its clarity.**

- Why do you always study the statistical properties of your least squares solution with high SNR only? What do you get for SNR = 10 dB for instance?

**The second Taylor series expansion of the inverse matrix  $(\mathbf{I} + \mathbf{M})^{-1}$  is valid when the spectral radius of  $\mathbf{M}$  is lower than 1. The spectral radius depends on the perturbation noise variance. As the noise variance increases, for fixed exact data, the SNR decreases and the spectral radius grows, as well as the error introduced by the Taylor series expansion. We observe that for large SNR the Taylor series expansion allows us to use the derived expressions that calculate the bias and the variance. When the relative error of these predictions increase, below 40 dB for example, the spectral radius is getting closer and eventually going over 1. With SNR = 10 dB the Taylor series expansion of the inverse matrix introduces an error unacceptably large.**

- Give the values of the matrices you use for simulating data sets so that the reader can test what you have carried out.

**The first paragraphs of Section 4.2, and Fig. 4.1, were added to describe the conditions in which the simulations were conducted from the structured and unstructured EIV problems.**

Two Monte Carlo (MC) simulation studies were conducted to test the obtained bias and variance formulas. There are two sets of formulas. One set was obtained for the unstructured and uncorrelated EIV problem, and, the other set, for the structured and correlated EIV problem. The problem solved by the data-driven step input estimation method (3.21) belongs to the second set. The objective of the MC simulations is to compare the results of the bias and variance formulas in these two cases. The comparison is an evaluation of the effect that the structure and correlation have in the problem solution. To make a fair comparison, the same exact data and the perturbation variance is used in both types of EIV problems.

The exact data is the system of linear equations (3.15) built by the data-driven step input estimation method. To build the system of equations, the processed signal  $y$  is the exact response of a sensor to the input  $0.1s$ , where  $s$  is the unit step response. The sensor is

modeled as a linear time-invariant mass-spring-damper system of order  $n = 2$ , that is given by

$$\mathbf{x}(k+1) = \begin{bmatrix} 0 & 1 \\ \frac{-k_s}{m+M} & \frac{-k_d}{m+M} \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ -g \end{bmatrix} u(k), \quad y(k) = [-1 \quad 0] \mathbf{x}(k), \quad (4.38)$$

where  $m = 0.015$  kg,  $k_s = 500$  kg/m,  $k_d = 0.5$  kg s/m,  $M = 0.1$  kg, and  $g = 9.81$  m/s<sup>2</sup>. The exact sensor response is shown in Figure 4.1.

In the simulations, the sample size considered is  $N = 200$ , since there the transient regime of the sensor model is still evident. In the case of the structured EIV problem, this sample size satisfies the requirements of the data-driven step input estimation method.

- Please remove Fig. 4.1 b from Section 4.2.1 because the top and bottom curves are obtained with different systems and simulation condition. Or explain why you can compare both. It could be interesting to test each time (i.e., in Section 4.2.1 and 4.2.2) the bias and covariance formulas obtained by considering the structured and unstructured EIV problems.

**Fig. 4.1 b has been relocated in its corresponding section. Now it is Fig. 4.7.**

- When you generate different simulation examples, explain why the new ones are interesting or, more importantly, what they bring w.r.t. the former simulations results. This is the case for instance when I compare Section 4.2.2 and Section 5.1.

**The first paragraph of Section 5.1 has been added to explain the purpose of the simulations in Chapters 5, with respect to the simulations in Chapter 4.**

The simulations described in Chapter 4 were conducted with an illustrative example that is assumed to be a second order weighing sensor. A weighing sensor was implemented to relate experimental results with the simulations. As it will be detailed later, the order of the practical sensor is at least is fifth order. Thus, the simulations presented in this section are useful to get insight on the impact of that increasing the model order has on the accuracy of the bias and variance predictions.

- Except if you can convince me, do not use the name "maximum likelihood" when you do not introduce a likelihood you want to maximize. In Chapter 6, you introduce a nonlinear least squares algorithm, nothing else (see the attached document, Numerical Optimization, Chapter 10).

**The method description in Section 6.2.2 has been modified to explain why the method is a maximum likelihood method. The modified paragraphs are as follows:**

Using a model of the sensor, the maximum-likelihood ML method simultaneously estimates the parameters of the applied affine input  $u(t) = at + b$ , and some of the sensor model parameters, using the observed sensor response  $\tilde{\mathbf{y}}$ . This method fits iteratively the response of the model  $\hat{\mathbf{y}}(\boldsymbol{\theta})$  to the sensor response by searching for the parameters  $\boldsymbol{\theta}$  that reduce the error difference between both responses. Algorithm 6.1 describes the formulation of an objective function  $f = \mathbf{r}^\top \mathbf{r}$ , where  $\mathbf{r}$  is the residual  $\mathbf{r}(\boldsymbol{\theta}) = \tilde{\mathbf{y}} - \hat{\mathbf{y}}(\boldsymbol{\theta})$ , that is minimized through the iterations. The objective function is the sum of the squares of the differences between the samples of both responses. Since the measurement noise is assumed Gaussian distributed with zero mean and variance  $\sigma_\epsilon^2$ , where the samples of this perturbation are independent and

identically distributed, the minimization of the objective function  $f$  maximizes the likelihood of fitting the actual sensor response with the model response.

The Jacobian matrix of the residual  $\mathbf{J}\boldsymbol{\theta} = \partial \mathbf{r}(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$  can be analytically derived using the sensor model. With the Jacobian matrix, the ML method searches the direction in which the residual decreases towards a local minimum. Unfortunately, this ML method cannot guarantee the global minimum since it depends strongly on the initialization of the optimization parameters, as it is discussed in [48]. Therefore, it is proposed to initialize the to-be-optimized parameters  $a$  and  $b$  using the results of the subspace method obtained after a convenient number of samples so that the affine input parameters are set close to the optimal values. The initialization of the remaining model parameters under study can be done using the sensor model representation. An illustrative example of the ML will be provided in a posterior subsection.

## Nikos Deligiannis

- Comments related to the structure of the thesis:

- In the introduction chapter it is best to describe a set of applications and settings that motivate your research and proposed solutions. This can make your thesis more accessible. Furthermore, you can add a subsection that describes in a high-level manner the remainder and structure of the thesis (describing briefly the contributions of each chapter). You can also mention your contributions there but in a less technical way since the problem statement only comes in the Chapter preliminaries.

**The motivation of the research has been improved in the introduction by describing real life examples of measurements in which the tradeoff between speed and estimation accuracy is noticeable. The structure of the thesis description and the contributions have also been revised in the Introduction.**

- In each main chapter (Chapter 4, Chapter 5, Chapter 6) you can add an introduction subsection, a state-of-the-art subsection and then elaborate your contributions and methods.

**The introduction of the chapters now includes the state of the art to position the PhD work with respect to the literature.**

- I believe that this structure will improve the understanding of the thesis and light up your contributions.

**Indeed, this structure allows to visualize the thesis contributions.**

- Further comments:

- Please discuss the complexity and scalability issues related to your approach in relation to the state of the art.

**The following paragraph has been added in Section 3.2 to discuss the scalability of the data-driven step input estimation method.**

The computational complexity of the RLS algorithm solution to the system of equations (3.15) is  $O\left((n+1)^2\right)$ . The largest computational requirement is for the initialization of the algorithm, where a matrix inverse is required when the number of samples is just enough to have a square matrix  $\hat{\mathbf{K}}_{n+1}$ . From there on, the RLS algorithm updates

the solution with linear complexity. Therefore, even though a large number of samples  $N$  makes the matrix in the system of equations (3.14) increase in the number of rows, the estimation of the solution does not increase in complexity. The data-driven step input estimation method is then scalable for any sensor of order  $n$ , and can be executed in devices with limited computational resources, provided that the computation of the  $n + 1 \times n + 1$  inverse matrix is feasible.

- Position your work with regard to the state of the art as well as regarding solutions based on low-rank matrix completion and compressed sensing.

**The last paragraph in Chapter 3 was added to describe the position the data-driven step input estimation method with respect to the most powerful unfalsified model of the behavioral approach theory. In the future work section, it was added a description of the need for efficient optimization algorithms that could solve the low-rank approximation problem enabling the online estimation of the step input value.**

- I recommend to use the formulation “we propose” than “I propose”

**The formulations similar to “I propose” were replaced by equivalents such as “we propose”.**

- You might wish to add a list of notation in the beginning of the thesis.

**A nomenclature list is provided before the summary of the document.**

## Philippe Dreesen

- General remarks:

- It would be good to include some layman’s examples that motivate the work of the thesis already in the introduction. In the current text, there is a nice example on page 52, almost at the end of the thesis. These kinds of examples should appear in the introduction to motivate and position the work.

**The following paragraphs have been added to the Introduction:**

Take for example the case of a thermometer. To measure temperature, the thermometer is put in contact with the substance or object of interest. Accordingly to the Newton’s law of cooling, the temperature  $y$  of the thermometer changes proportionally to the its difference with respect to the temperature  $u$  of the matter under study:

$$\frac{dy}{dt} = k_T (u - y) \quad (2.1)$$

where the proportionality constant  $k_T$  is a parameter of the thermometer first order model. The input  $u$  is unknown and its estimation is the objective of the measurement. It is customary to take the reading from the thermometer once the temperature stops changing.

Another popular example of a measurement is found in weighing. In most of modern scales, the object of interest is placed on top of a weighing sensor. A simple model of the weighing sensor is the second order mass-spring-damper model shown in Figure 2.1. In this model, the unknown mass  $u$  causes a displacement  $y$  of the scale and the dynamics of the sensor are described by the differential equation:

$$\frac{d}{dt} \left( u(t) \frac{dy}{dt} \right) + k_d \frac{dy}{dt} + k_s y = u(t) \quad g \quad (2.2)$$

where  $m$  is the own mass of the scale,  $k_d$  is the damping constant,  $k_s$  is the elasticity constant, and  $g = 9.81 \text{ m/s}^2$  is the gravitational acceleration.

For illustration purposes, Figures 2.2 and 2.3 show the responses of the thermometer and the weighing sensors, respectively, after the application of a constant input  $u$ , which can be represented by a step input.

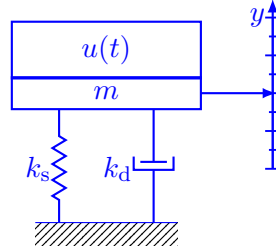


Figure 2.1: A second order mass-spring-damper model represents the weighing sensor. The application of the mass  $u$  causes a change in the position  $y$  of the scale.

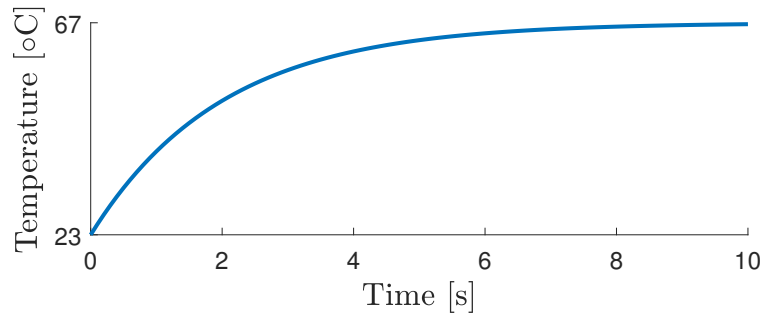


Figure 2.2: This simulation shows an example of the temperature change experimented by a thermometer, with  $k_T = 0.5 \text{ s}^{-1}$ , caused by applying a step input of  $67 \text{ }^\circ\text{C}$  when the thermometer was initially at  $23 \text{ }^\circ\text{C}$ . In this example, we should wait  $10 \text{ s}$  to have an estimation of the input with a relative error lower than  $0.5\%$ .

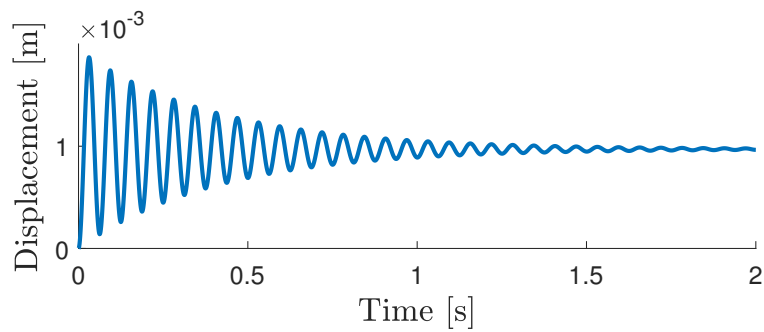


Figure 2.3: This simulation illustrates the displacement observed when a constant input  $u = 1 \text{ kg}$  is applied to a weighing sensor in equilibrium. The sensor parameters are  $k_d = 5 \text{ Ns/m}$ , and  $k_s = 10000 \text{ N/m}$ . In this conditions, the user has to wait more that  $2 \text{ s}$  to observe a stabilization on the scale and to estimate the mass from reading the final value of the displacement with a relative error smaller than  $0.72\%$ .

- Detailed remarks:

**The detailed remarks in this list have been attended accordingly:**

- Citations that are used in-line should have brackets only around the year, not around the authors' names. For instance on page 3, line -2 is written “reviewed in [Hack and ten Caten, 2012]”. This should be displayed as “reviewed in Hack and ten Caten [2012]. In LaTeX this can typically be done by making the proper distinction between the commands `citet` and `citep`. The former is used for references in text, the latter for parenthetical references.
- On page 1 of the PDF (front of thesis), I believe also N. Deligiannis should be mentioned with the title “prof.”.
- Page 1 of thesis: typo on line 6: “and on the sensor initial conditions” =<sub>i</sub> “sensor’s initial conditions”
- Page 3, line -2: “uncertainty analysis are reviewed” =<sub>i</sub> “is reviewed”
- Page 3, line -1: remove comma before “that”
- Page 3, line -2: fix the in-text reference (see above)
- Page 4, lines 11-13: fix the in-text references (see above)
- Page 5, line 5: Remove “The” at the beginning of the sentence (The Hankel...)
- Page 6, line 10: fix the in-text references (see above)
- Page 7, lines 10-11: the sentence on heart and breathing physiological monitoring sounds a bit awkward. Please reformulate.
- Page 9, line -1: fix the in-line reference (see above)
- Page 10, lines 1-3: in the enumeration of the types of sensors, remove the article ‘the’ before the sensor name. So: “like three-axis sensors [ref], radio-frequency intruder sensors [ref], and radar sensors [ref].
- Page 10, equation 3.10: The symbol  $D$  was not bold in the equations above. Be consistent!
- Page 10, caption of figure: replace “reverts” by “inverts”
- Page 11, line 9, put “poles” between parentheses instead of between commas.
- Page 12, line -6: “that in matrix form is” =<sub>i</sub> “which in matrix form is”
- Page 13, line -9: “in the practice” =<sub>i</sub> “in practice”
- Page 14, line -14: missing space between “problem” and “(3.11)”
- Page 16, halfway page: the sentence “Applying a second-order...” is not a proper sentence (perhaps fix this by removing “that”).
- Page 16 introduces seemingly some new notation, e.g.,  $b(\cdot)$ ,  $\mu(\cdot)$  and  $C(\cdot)$ . While their meaning is clear from the context, it is desirable to give proper definitions when introducing new mathematical tools.
- Page 17, line -1: fix in-text references (see above)
- Page 20 (and beyond): The abbreviation CRLB is introduced, but sometimes CRB is used instead.
- Page 21, line -10: fix in-text reference (See above)

- Page 35, second sentence “The sensor is a dynamic...” is not a proper sentence.
- Page 50: Try to reorganize this section into two or more paragraphs.
- Page 59, line 2: fix in-text reference (See above)
- Page 74, line 11: “The learning obtained from...” this is an awkward sentence; please reformulate
- Page 81, all three journal publication references do not have (or have wrong?) page numbers.

## References

- [1] J. W. Neuberger, The continuous Newton’s method, inverse functions, and Nash-Moser, *The American Mathematical Monthly* 114 (2007) 432–437.
- [2] G. Balkema, P. Embrechts, Linear regression for heavy tails, *Risks* 6 (3) (2018) 93. doi:10.3390/risks6030093.
- [3] R. A. Maronna, R. D. Martin, V. J. Yohai, *Robust statistics: theory and methods*, John Wiley & Sons, West Sussex, UK, 2006.
- [4] Z. Zhang, Parameter estimation techniques: a tutorial with application to conic fitting, *Image and Vision Computing* 15 (1) (1997) 59 – 76. doi:10.1016/S0262-8856(96)01112-2.
- [5] P. J. Huber, *Robust statistics*, John Wiley & Sons, Cambridge, MA, 1981. doi:10.1002/0471725250.
- [6] I. Markovsky, S. Van Huffel, Overview of total least-squares methods, *Signal Processing* 87 (10) (2007) 2283–2302. doi:10.1016/j.sigpro.2007.04.004.
- [7] S. Van Huffel, C. Cheng, N. Mastronardi, C. Paige, A. Kukush, Total least squares and errors-in-variables modeling, *Computational Statistics & Data Analysis* 52 (2) (2007) 1076–1079. doi:10.1016/j.csda.2007.07.001.
- [8] A. Beck, Y. C. Eldar, Structured total maximum likelihood: an alternative to structured total least squares, *SIAM Journal on Matrix Analysis and Applications* 31 (5) (2010) 2623–2649. doi:10.1137/090756338.
- [9] T. Söderström, *Errors-in-variables methods in system identification*, Springer, 2018. doi:10.1007/978-3-319-75001-9.
- [10] S. Rhode, K. Usevich, I. Markovsky, F. Gauterin, A recursive restricted total least-squares algorithm, *IEEE Transactions on Signal Processing* 62 (21) (2014) 5652–5662. doi:10.1109/TSP.2014.2350959.
- [11] S. Rhode, F. Bleimund, F. Gauterin, Recursive generalized total least squares with noise covariance estimation, *IFAC Proceedings Volumes* 47 (3) (2014) 4637 – 4643. doi:10.3182/20140824-6-ZA-1003.01568.
- [12] A. Djouambi, A. Voda, A. Charef, Recursive prediction error identification of fractional order models, *Communications in nonlinear science and numerical simulation* 17 (6) (2012) 2517 – 2524. doi:doi.org/10.1016/j.cnsns.2011.08.015.

- [13] L. Shang, J. Liu, Y. Zhang, Recursive fault detection and identification for time-varying processes, *Industrial & Engineering Chemistry Research* 55 (46) (2016) 12149–12160. doi:10.1021/acs.iecr.6b02653.
- [14] P. Gil, F. Santos, L. Palma, A. Cardoso, Recursive subspace system identification for parametric fault detection in nonlinear systems, *Applied Soft Computing* 37 (2015) 444 – 455. doi:10.1016/j.asoc.2015.08.036.
- [15] J. A. Gubner, *Probability and random processes for electrical and computer engineers*, Cambridge University Press, 2006. doi:10.1017/CB09780511813610.
- [16] J. O. Smith, *Spectral audio signal processing*, W3K publishing, 2011. URL <https://ccrma.stanford.edu/~jos/sasp/>
- [17] B. Neal, On the bias-variance tradeoff: textbooks need an update (2019). arXiv:1912.08286.
- [18] S. Geman, E. Bienenstock, R. Doursat, Neural networks and the bias/variance dilemma, *Neural Computation* 4 (1) (1992) 1–58. doi:10.1162/neco.1992.4.1.1.
- [19] Manning, C. D. and Raghavan, P. and Schütze, H., *Introduction to information retrieval*, Cambridge University Press, Cambridge, UK, 2008.
- [20] T. Hastie, R. Tibshirani, J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, 2nd Edition, Springer, New York, 2009. doi:10.1007/978-0-387-84858-7.
- [21] G. James, D. Witten, T. Hastie, R. Tibshirani, *Linear model selection and regularization*, Springer, New York, NY, 2013, pp. 203–264. doi:10.1007/978-1-4614-7138-7\_6.
- [22] S. Arlot, A. Celisse, A survey of cross-validation procedures for model selection, *Statist. Surv.* 4 (2010) 40–79. doi:10.1214/09-SS054.
- [23] H. Akaike, A new look at the statistical model identification, *IEEE Transactions on Automatic Control* 19 (6) (1974) 716–723. doi:10.1109/TAC.1974.1100705.
- [24] J. Rissanen, Modeling by shortest data description, *Automatica* 14 (5) (1978) 465 – 471. doi:10.1016/0005-1098(78)90005-5.
- [25] R. McElreath, *Statistical rethinking, a Bayesian course with examples in R and STAN*, 2nd Edition, CRC press, Boca Raton, FL, 2020.
- [26] Y. Eldar, G. Kutyniok, *Compressed sensing: theory and applications*, Cambridge university press, 2012.
- [27] M. Rani, S. B. Dhok, R. B. Deshmukh, A systematic review of compressive sensing: concepts, implementations and applications, *IEEE Access* 6 (2018) 4875–4894.
- [28] D. Craven, B. McGinley, L. Kilmartin, M. Glavin, E. Jones, Compressed sensing for bioelectric signals: a review, *IEEE Journal of Biomedical and Health Informatics* 19 (2) (2015) 529–540. doi:10.1109/JBHI.2014.2327194.
- [29] M. F. Duarte, Y. C. Eldar, Structured compressed sensing: from theory to applications, *IEEE Transactions on Signal Processing* 59 (9) (2011) 4053–4085.



- [30] L. Kezhi, C. Shuang, State of the art and prospects of structured sensing matrices in compressed sensing, *IEEE Access* 9 (5) (2015) 665–677. doi:10.1007/s11704-015-3326-8.
- [31] J. Schmidhuber, Deep learning in neural networks: an overview, *Neural Networks* 61 (2015) 85 – 117. doi:10.1016/j.neunet.2014.09.003.
- [32] S. Hosseinyalamdary, Deep Kalman filter: simultaneous multi-sensor integration and modelling; a GNSS/IMU case study, *Sensors* 18 (5) (2018) 1–15. doi:10.3390/s18051316.
- [33] J. R. Rambach, A. Tewari, A. Pagani, S. D., Learning to fuse: a deep learning approach to visual-inertial camera pose estimation, in: 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2016, pp. 71–76.
- [34] Z. Zheng, H. Chen, X. Luo, A Kalman filter-based bottom-up approach for household short-term load forecast, *Applied Energy* 250 (2019) 882 – 894. doi:10.1016/j.apenergy.2019.05.102.
- [35] Y. Bai, X. Wang, X. Jin, Z. Zhao, B. Zhang, A neuron-based Kalman filter with nonlinear autoregressive model, *Sensors* 20 (1) (2020) 299. doi:10.3390/s20010299.
- [36] S. Farrell, P. Calafiura, M. Mudigonda, Prabhat, D. Anderson, J. Vlimant, S. Zheng, J. Bendavid, S. M., G. Cerati, G. L., K. J., P. Spentzouris, A. Tsaris, Novel deep learning methods for track reconstruction (2018). arXiv:1810.06111.
- [37] S. Shashua, S. Mannor, Deep robust Kalman filter (2017). arXiv:1703.02310.
- [38] R. G. Krishnan, U. Shalit, D. Sontag, Deep Kalman filters (2015). arXiv:1511.05121.
- [39] J. C. Willems, From time series to linear system - part i. finite dimensional linear time invariant systems, *Automatica* 22 (1986) 561–580. doi:10.1016/0005-1098(86)90066-X.
- [40] J. Willems, P. Rapisarda, I. Markovsky, B. De Moor, A note on persistency of excitation, *Systems & Control Letters* 54 (4) (2005) 325 – 329. doi:10.1016/j.sysconle.2004.09.003.
- [41] I. Markovsky, The most powerful unfalsified model for data with missing values, *Systems & Control Letters* 95 (2016) 53 – 61. doi:10.1016/j.sysconle.2015.12.012.
- [42] I. Markovsky, K. Usevich, Structured low-rank approximation with missing data, *SIAM Journal on Matrix Analysis and Applications* 34 (2) (2013) 814–830. doi:10.1137/120883050.
- [43] I. Markovsky, K. Usevich, Software for weighted structured low-rank approximation, *Journal of Computational and Applied Mathematics* 256 (2014) 278 – 292. doi:10.1016/j.cam.2013.07.048.
- [44] J. Ying, J. Cai, D. Guo, G. Tang, Z. Chen, X. Qu, Vandermonde factorization of Hankel matrix for complex exponential signal recovery-application in fast nmr spectroscopy, *IEEE Transactions on Signal Processing* 66 (21) (2018) 5520–5533. doi:10.1109/TSP.2018.2869122.
- [45] J. Hou, L. Chau, N. Magnenat-Thalmann, Y. He, Sparse low-rank matrix approximation for data compression, *IEEE Transactions on Circuits and Systems for Video Technology* 27 (5) (2017) 1043–1054. doi:10.1109/TCSVT.2015.2513698.
- [46] R. Pintelon, J. Schoukens, System identification: a frequency domain approach, 2nd Edition, IEEE Press, Piscataway, NJ, 2012.

- [47] T. Kailath, H. Sayed, B. Hassibi, Linear estimation, 1st Edition, Prentice Hall, NJ, 2000.
- [48] J. Nocedal, S. Wright, Numerical optimization, 2nd Edition, Springer Verlag, New York, 2006.