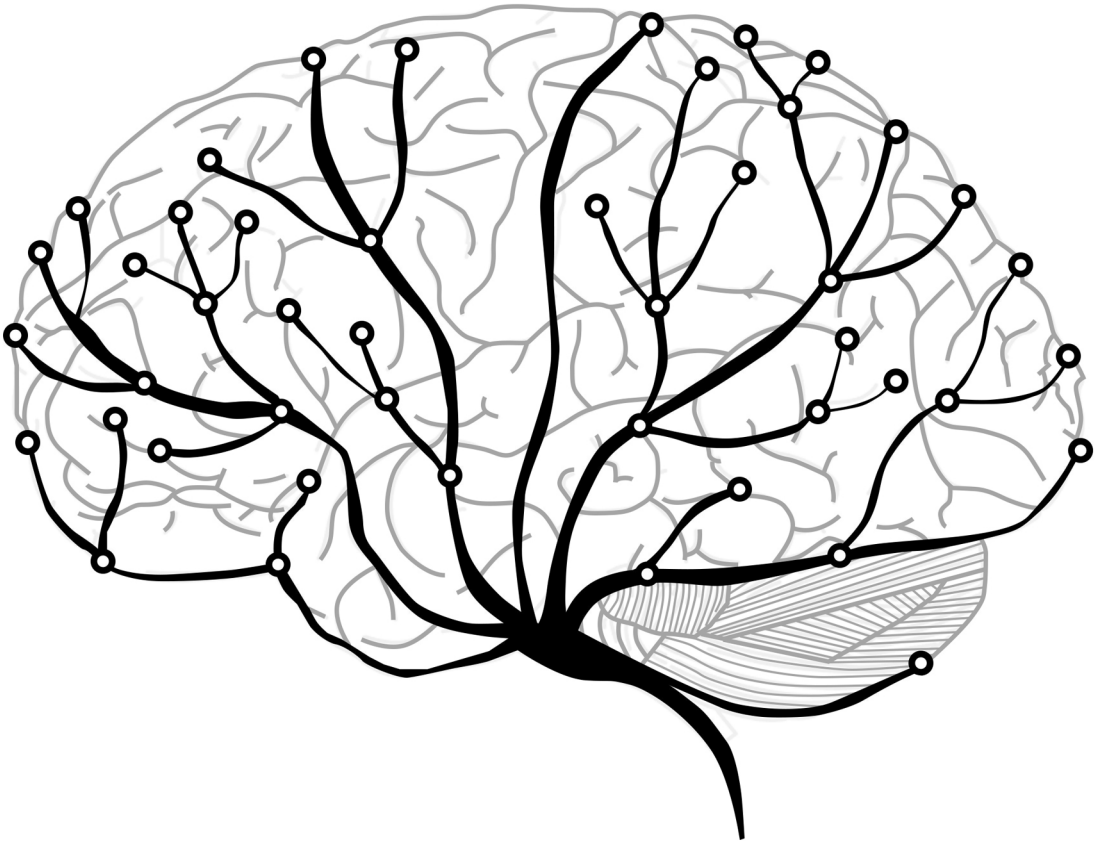
Bayesian Deep Learning for Dynamic System Identification

Ibrahim Chahine

Master of Science Thesis



TU Delft Delft University of Technology

Delft Center for Systems and Control

Bayesian Deep Learning for Dynamic System Identification

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Ibrahim Chahine

March 19, 2021

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported and supervised by faculty members of the Cognitive Robotics Department (CoR) and the Delft Center for Systems and Control (DCSC). The contribution and cooperation of Dr. Wei Pan and Phd Student Honpeng Zhou and Dr. Jens Kober is gratefully acknowledged.

Abstract

System identification is a mature field in physical sciences and an emerging field in social sciences, with a vast range of applications. Nevertheless, it remains of great focus in academia. The main challenge is the efficient use of data to generate good model fits. System identification involves multi-disciplinary techniques from statistical, mathematical and computational sciences. The typical approaches for dynamic system identification include fuzzy models, non-linear auto regressive models, state-space models, subspace identification models and many others. In this thesis, artificial neural networks are evaluated, among these, as black-box methods known to be capable of universal approximation. With no essential prior information, the identification problem exhibits more difficult challenges. These include the complexity of the resulting models, choice of regressors, and uncertainty quantification. Specifically in this thesis, a Sparse Bayesian Learning approach is proposed, as a solution to these challenges. A practical iterative Bayesian procedure is derived and set to identify six benchmark datasets of three non-linear mechanical processes: Cascaded Tanks, Coupled Electric Drives, Bouc-Wen hysteresis model as well as of three linear mechanical processes: Heat Exchanger, Glass Tube Manufacturing and Hair Dryer.

Preface

This document acts as a tentative fulfillment of my Master of Science graduation thesis in Systems and Control at TU Delft.

I would like to deeply thank my supervisors Dr. Wei Pan and Hongpeng Zhou for their assistance during the writing of this thesis. Thank you Dr. Jens Kober for allowing me to take on this opportunity.

I would like to dedicate my work to my partner and my family, who stood by me all along.

Table of Contents

Preface iii

1 Introduction 3

2 Prior Context 5

2-1 Dynamic System Identification: An overview . . . 6

2-1-1 The Identification Setup . . . 6

2-1-2 Challenges of System Identification . . . 7

2-1-3 Neural Networks: A Black-Box Approach . . . 9

2-2 Sparse Bayesian Learning . . . 12

2-2-1 Occam's razor and the Marginal Likelihood . . . 12

2-2-2 Inference and Prediction . . . 14

2-2-3 Likelihood and Sparsity Inducing Priors . . . 14

2-2-4 Development of an Optimization Problem . . . 16

2-2-5 Bayesian Deep Learning . . . 18

3 Manuscript 21

4 Conclusion 47

4-1 Summary . . . 47

4-2 Limitations . . . 47

4-3 Opportunities . . . 48

Bibliography 49

Appendix **55**

4-4 Appendix A: Benchmarks Description 56

4-4-1 Heat Exchanger 56

4-4-2 Glass Tube Manufacturing 57

4-4-3 Hair Dryer 60

4-4-4 Cascaded Tanks 62

4-4-5 Coupled Electric Drives 65

4-4-6 Bouc-Wen Hysteresis Model 67

List of Figures

2-1 Bayesian Learning in the System Identification Cycle. 6

2-2 Weight matrix between layer $l-1$ and l and between layer l and $l+1$. Courtesy of Dr. Wei Pan [47] 10

2-3 Layer l of a NN with regularized groups in red. Courtesy of Dr. Wei Pan [47] . . 10

2-4 Approximations of the prior and posterior probability. Reprinted by permission from Springer Nature: Springer, Maximum Entropy and Bayesian Methods, Chapter 3 [33] by MacKay, David J. C. © 1992 13

2-5 Comparison between full and approximate prior models equiprobability contours with the likelihood. From Wipf article "Perspectives on Sparse Bayesian Learning" [71] in IEEE Transactions on Signal Processing, vol. 52, no. 8, pp. 2160 © 2004 IEEE 17

2-6 Figure showing the conventional and Bayesian views of a perceptron. 18

4-1 Heat Exchanger estimation input data. 57

4-2 Heat Exchanger validation input data. 57

4-3 Glass Tube Manufacturing Process [68]. This article was published in the 10th Triennial IFAC Congress on Automatic Control, Volume 20, V. Wertz and G. Bastin and M. Haest, Identification of a glass tube drawing bench, Page 334, © Elsevier (1987). 58

4-4 Glass Tube Manufacturing estimation input data. 59

4-5 Glass Tube Manufacturing validation input data. 59

4-6 PT 326 Process Trainer Sketch 60

4-7 Hair Dryer estimation input data. 61

4-8 Hair Dryer validation input data. 61

4-9 Sketch of the Cascaded Tanks system 63

4-10 Cascaded Tanks estimation input data. 63

4-11 Cascaded Tanks validation input data. 63

4-12 Coupled Electric Drive Sketch 65

4-13 First CED provided input data. 65
4-14 Second CED provided input data. 65
4-15 Bouc-Wen estimation input data. 68
4-16 Multisine Validation Input for BW Benchmark. 68
4-17 Sinesweep Validation Input for BW Benchmark. 68

List of Tables

4-1 Comparison of other models from literature on Heat Exchanger Dataset 57
4-2 Comparison of other models from literature on Glass Tube Manufacturing Dataset 59
4-3 Comparison of other models from literature on Hair Dryer Dataset 62
4-4 Comparison of other models from literature for Cascaded Tanks Benchmark . . . 64
4-5 Comparison of other models from literature for CED Benchmark 66
4-6 Comparison of other models from literature for Bouc-Wen Benchmark 69

"We are like dwarfs sitting on the shoulders of giants. We see more, and things that are more distant, than they did, not because our sight is superior or because we are taller than they, but because they raise us up, and by their great stature add to ours."

— *John of Salisbury*

Chapter 1

Introduction

The identification of control processes is an important step into transferring essential knowledge to intelligent automated systems. With the complexity of dynamic systems comes the inefficiency in rule-based processes to deal with contingencies in terms of prediction or control. To this end, many new methods have emerged combining linear and non-linear systems theory and new data mining techniques. Neural networks are non-linear black box models with a great potential for model fitting. The application of neural networks on system identification was first suggested in 1990 [37]. Until the present day, multiple textbooks emerged treating the subject for both control and system identification [23, 48, 72].

Using neural networks for system identification raises some challenges. Neural networks exhibit a universal approximative capability under mild conditions [15], however they can render over-parameterized models to the process it represents and thus degrade model generalization properties. It is, then, important to realize that overfitting and complexity are issues that needs to be addressed [61], specifically with noisy and short measurements. In addition to that, the use of a Neural Network as the identification model abolishes the need to select a basis function space especially with the identification of non-linear systems, but the structure and regressors are hardly a given. Finally, neural networks have been long criticized for their opacity and has often been mystified for its complex structure, but also because its parameters do not represent any physical quantity.

These challenges are addressed, inspired by a Bayesian perspective on neural networks. By modeling the neural network in a Bayesian approach, one can reduce model redundancies and ease overfitting by using sparsity-inducing priors and pruning on model parameters [74]. One can also quantify uncertainty in all inferences, which is primarily useful for decision making in safety critical applications [17, 36].

There exist a large variety of Bayesian learning applications to system identification that differ in the Bayesian treatment (Laplace approximation, Expectation Maximization, Monte Carlo methods, ...), the model structure (neural networks, fuzzy models, ARX, ...) and the applications. What this thesis intends to evaluate, is Sparse Bayesian Supervised Learning for the identification of various mechanical processes, using two types of neural networks (MLP and LSTM) as the model structure and the Laplace approximation as the Bayesian approach.

An algorithm is developed and tested for three benchmark datasets of non-linear systems provided on the web page http:/www.nonlinearbenchmark.org/ and three linear systems benchmark data found in Matlab system identification toolbox https://nl.mathworks.com/help/ident/examples.html.

This thesis contributions can be summarized as follows:

- Derivation of a practical identification algorithm using Sparse Bayesian Learning theory on Neural Networks.
- Introducing, using a Bayesian formulation of regularization, the concept of structured sparsity that leads to sparse compact models.
- Demonstrate these derived properties (sparsity, uncertainty and competitive free run simulation performance) by identifying mechanical systems using benchmark data with a comparison with previous works.

The document is organized as follows. Chapter 2 helps placing this work in its literature context. Chapter 3 is the manuscript that represents the body of the thesis. Chapter 4 concludes the work with a small discussion on limitations and opportunities.

Chapter 2

Prior Context

This chapter aims to summarize challenges of the identification of linear/non-linear systems and presents an overview of the opportunities and new paradigms treating Neural Networks and Bayesian Learning.

2-1 Dynamic System Identification: An overview

2-1-1 The Identification Setup

System identification is an iterative process that is often described as a loop [32]. It involves mainly modeling and design choices [67] such as type of models, basis functions, parameter update, choice of input-output pairs and finally a quality assessment method (validation).

Any system identification framework relies on measurement data and/or model prior information. Briefly, an experiment is designed, to acquire crucial data for model fitting. This mainly includes input signal frequency and amplitude content. For linear models, it is important that the signal used for estimation excites a large range of frequencies. However for non-linear models, both signal frequency and amplitude content are relevant.
A model structure is chosen and the model is optimized to fit the data. The resulting model is validated with the validation data according to a figure of merit. If the model is deemed satisfactory, it is accepted, else design choices related to the data or model used are revisited.

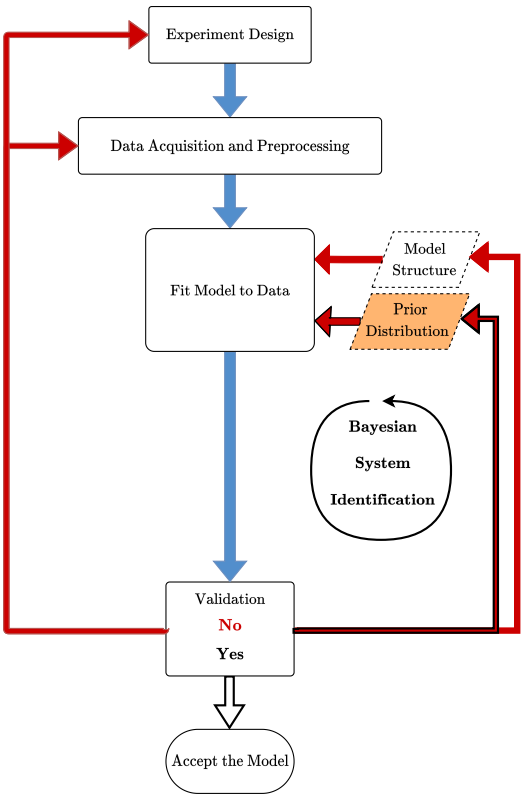


Figure 2-1: Bayesian Learning in the System Identification Cycle.

With a Bayesian approach to system identification discussed in this thesis, the choice of priors is crucial to the development of an optimization procedure and can be revisited similarly to other design choices in the system identification cycle. The diagram in figure 2-1 shows the system identification cycle and the introduced iterative Bayesian approach.

In this thesis, the data used for system identification is taken from benchmarks of dynamic systems in the case of non-linear system identification and Matlab examples in the case of linear system identification. A more thorough description of these is provided in chapter 4-4. Data is pre-processed if needed and fed into the Bayesian inference cycle.

2-1-2 Challenges of System Identification

Both linear and non-linear dynamic system identification challenges are well described in literature. Multiple modeling approaches to these challenges are adopted. Challenges to dynamic system identification include:

- Lack of Data and difficulties in **data acquisition** or measurement: The availability of good data is crucial to good model estimation. Some issues related to data acquisition include bad signal to noise ratio, inputs that cannot be maneuvered by nature, a slow time scale to a process and unmeasurable disturbances [31].
- **High dimensionality** and the right choice of a low dimensional coordinate system that allows an acceptable representation of the process. This remains a subject of study with notable contributions such as Eigensystem Realization Algorithm and Dynamic Mode Decomposition. Dimensionality is also related to the selection of **model order** which determines the regressors chosen to fit the data.
- Success in system identification lies in insights and intuition into the problem. The problem is application dependent and an attempt to generalize by automating the model construction procedure can prove very difficult [31].

However it is widely common to see more challenges related to non-linear processes than to linear ones. These challenges are best summarized below:

- Non-linear dynamic systems are often distinguished from linear ones by the use of the **superposition principle**. It states that the resulting response of multiple inputs is the sum of individual responses to each of these inputs. The following principle is not satisfied by non-linear systems, resulting in difficulties in system identification. Linear approximations often fail, and are reliable when control actions span smaller windows around linearization [8]. Additionally, violation of the superposition principle implies a need to evaluate the model for different input amplitudes, as the response to distinct step inputs are not linearly deducible. This means non-linear systems need a much higher amount of plant tests with varying amplitude and frequency signals [8].
- Given noisy time series observations, non-linear system identification aims to recover the non-linear set of equations that can describe a given process. In some cases, the complexity and coupling in states of these systems are too complex to model and sometimes not measurable, such as hysteresis and backlash [42]. In other cases, mathematical modeling fails to provide approximations adequate enough to represent the process. With this, comes the **challenge of specifying the model structure**. Approaches to identifications can be categorized by the following model classes: Volterra series models, Fuzzy systems, Neural Networks, NARMAX models, block-structured models, stochastic state-space representations and others. [47].

In other words, for black-box or grey-box approaches, the non-linear basis function candidates selection is a crucial design step for identification. This selection can be done with prior knowledge of the dynamics or physical insight, but also can include a infinite dictionary of functions in theory, for instance by the usage of universal approximators like neural networks and fuzzy models [30].

- **Identifiability** is a very central property in system identification. It is the ability of the chosen set of candidate functions to be uniquely determined from provided data. Assuming the following simple identification problem:

$$\hat{w} = \arg\min \ ||y - \Phi w||_2^2 \tag{2-1}$$

Φ being the chosen candidate functions, y the output and w the weights associated with the candidates. Global identifiability requires a global unique solution $\hat{w}$ over w in the space of models D_M. Local identifiability, on the other hand, is achieved if the optimal $\hat{w}$ is constrained to a small neighborhood of w. With the choice of non-linear candidate functions and the available data, identifiability is hardly provided in non-linear systems [47]. With this comes the problem of **uniqueness** of the solution obtained. Identifiability is thus hardly given, and dangerous to assume [47].

- The design of the model, whether it is non-linear basis function space, or neural networks or fuzzy systems, along with the input space dimensionality could result in an increased **model complexity** in system identification [42]. This complexity is reflected in the model validation phases, where **overfitting** may be noticed. When X in equation 2-1 is coherently chosen, one can reach a unique solution by enforcing sparsity over W. However such an assumption is hardly given. One way to enforce sparsity, is through **regularization**. The ill-posed problem defined in the previous bullet point can be recast into the following linear regression form [47]:

$$\hat{w} = \arg\min \ ||y - \Phi w||_2^2 + \lambda ||w||_{l_i} \tag{2-2}$$

It is important to note that λ is the regularization trade-off parameter and i in l_i defines the i-norm. This can involve l_0, l_1 (lasso regression) and l_2 (ridge regression) norms and indicate different levels of sparsity in relation to the objective function in equation 2-2 [7]. The l_0 norm is known to result in a computationally expensive, non-convex optimization. It is also known to be NP-hard [49]. The l_1 norm is often used as a convex relaxation to the l_0 norm. The main idea is that the l_1 norm is the convex envelope of the l_0 norm [47, 49].

- Non-linear system identification often involves **non-convex optimization problems** [47]. One of the methods used to solve such optimization problems is the *Convex Concave Procedure* (CCCP) [47]. Given a non convex function f that can be recast to the difference of convex functions u and v.

$$\min_{x \in C} f(x) = \min_{x \in C} u(x) - v(x) \tag{2-3}$$

where $C \subseteq \mathbb{R}^p$. This optimization can be formulated iteratively by the CCCP as a minimization of $\bar{f}$, the majorisation function of f evaluated at $x(k)$ [73].

$$x(k+1) = \arg\min_{x \in C} \bar{f}(x, x(k)) \tag{2-4}$$

$$\bar{f}(x, x(k)) = u(x) - x^T \nabla v(x(k)) \tag{2-5}$$

with v a differentiable function. The main idea is to linearize the concave function -v around the solution found at iteration k, then minimize equation 2-4. The algorithm can extend to handle constraints, and can converge to stationary points [27].

Other problems involve the **absence of analytical solutions**, **NP hardness**, where one refers to approximations and convex relaxations to obtain a solution [42, 47].

2-1-3 Neural Networks: A Black-Box Approach

Artificial Neural Networks (ANN) belong to the family of black-box models that attempt to imitate the human brain's networks of neurons as processing elements. Organized in layers, they interconnect forming a complex network. These processing units are called perceptrons. In supervised learning, the network is trained to minimize the difference between the model output and the desired output of the network, in other words, the prediction error. Several architectures exist for ANN and the application to system identification has been studied extensively. These include Multi-Layer Perceptron (MLP), Recurrent Neural Networks (RNN) and Radial Basis Networks (RBN). In this thesis, the subject of study is the use of MLPs and LSTMs within a Bayesian framework.

Approximative Ability: The approximation capabilities of feed-forward MLP has been long investigated in literature. It is found that MLP are universal approximators in theory. MLP networks are capable of representing a Borel-measurable continuous function in a given interval with one layer and a finite number of nodes up to a precision, given that the activation functions used are continuous and bounded [15, 29]. These are mild requirements and the sigmoidal activation function is a simple example. However, the number of neurons needed or the network structure is not a given [15]. A similar result was found in 1992 for RNNs in a paper proving that it is capable of simulating all Turing machines [55]. In 2006, specific to dynamic system identification, a paper by Schaffer et al. proved universal approximative abilities of RNN in state space model form [60].

Identifiability: It is challenging to prove identifiability, especially that the obtained NN model is rarely unique. However, one can propose that the sparest solution may be a unique one [47]. Penalizing model parameters and imposing sparsity induces a drop out on weights or neurons depending on the choice of the regularizer. This reduces model complexity and size and is analogous to the sparse distributed functions of neurons in brain. At a given time, only a small percentage of neuron pathways are activated when performing a function.

There are many different ways to regularize. In addition to the choice of regularization norm function already mentioned in the section 2-1-2, there exists different approaches to sparsity in NN. Considering perceptrons as processing units, one can enforce structural sparsity and compactness of networks by using group regularization [47, 53].

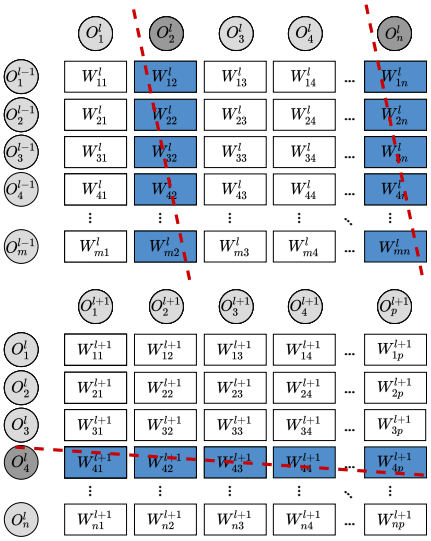


Figure 2-2: Weight matrix between layer $l-1$ and l and between layer l and $l+1$. Courtesy of Dr. Wei Pan [47]

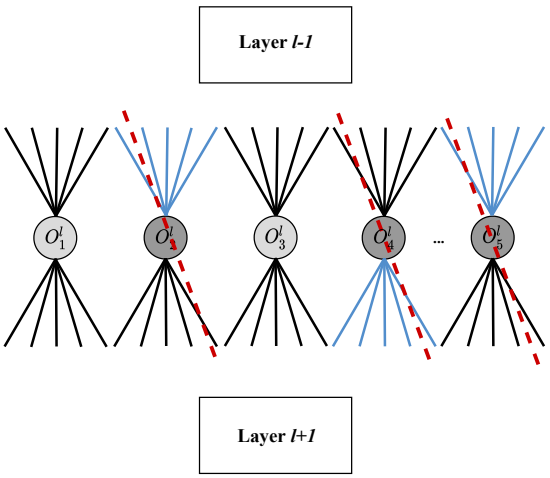


Figure 2-3: Layer l of a NN with regularized groups in red. Courtesy of Dr. Wei Pan [47]

Dropping these perceptrons is guided by dropping the group of connections to a neuron O_i^l from layer $l-1$. This is called in-group regularization and is shown in the weight matrix on top of figure 2-2. The same goes to the weight connections from neuron O_i^l to layer $l+1$. It is called out-group regularization and is shown in the bottom weight matrix of figure 2-2.

Model Estimation: Neural Networks are simple and easy to train, but most importantly they do not exhibit problems with dimensionality of inputs. Increasing the number of inputs only linearly increases the number of parameter with a fixed network architecture [41]. In addition, an uneven data distribution is not an issue as, neural networks can find, using the inner hidden transformations, the suitable coordinate system and directions of non-linearity (if it applies) [41].

Training the network is done by the use of stochastic gradient descent and is called backward propagation, where the output loss is used to adjust the hidden weights. Specific to system identification, the inputs are generally chosen to be k lagging elements of inputs u and output y, a measure of the prediction error $\hat{y}(t+1) - y_{true}(t+1)$ and the regularization term constituting the loss is differentiated with respect to all hidden parameters. This gradient is used to update the network's parameters through stochastic gradient descent given in the equation 2-6.

$$w(t+1) = w(t) - \eta \nabla_w L) \tag{2-6}$$

There exists many other variants and extensions of stochastic gradient descent for an adaptive update of parameters. Among them RMSProp and ADAM are very commonly used algorithms. RMSProp uses exponentially weighted averages of gradients with previous gradients to compute the learning rate for weights update [51]. ADAM, on the other hand, uses an exponentially weighted average of gradients and an exponentially weighted average second moments of gradients at each iteration for updating both the learning rate and weights [51].

There also exists second order methods for the parameters update. These capture better information than first order methods and thus improves search directions. Second order methods require a hessian matrix computation which is infeasible for Deep Neural Networks in terms of explicit calculation and storage [5]. However some very efficient approximations of the hessian in literature provide exciting prospects. Examples fall in the category of Inexact-Newton methods, Quasi-Newton methods, Gauss-Newton methods and Trust-Region methods [64].

One main criticism that Neural Networks have for dynamic system identification is that these are opaque and do not provide any physical intuition for the system in hand. They are constituted of hidden parameters, that do not represent any physical quantity. In the case of feed-forward neural networks, they can be thus seen as a set of non-linear static approximations to dynamic systems. These parameters are however accessible for the user. While many of the systems we study today can be accurately modeled using physical laws, many others are hardly/modestly modeled and some have highly coupled, hardly measurable or highly adaptive parameters. System identification, using black-box models do not provide complete physical intuition, but they demonstrate an excellent approximative ability of a reasonable governing function in the system. This field has gained lots of recent interest in the system identification community. Specific to the application of Neural Networks on modeling and system identification , one can find multiple textbooks [48, 72, 23].

2-2 Sparse Bayesian Learning

This chapter aims to provide an overview of Sparse Bayesian Learning. This section starts with Occam's razor and finishes with Bayesian Deep Learning. Incorporating system identification in a probabilistic framework has been an extensive subject of study for the last three decades. With a Bayesian approach, one can reformulate the optimization to more efficient expressions than standard regularization formulations such as Lasso-type algorithms [47].

2-2-1 Occam's razor and the Marginal Likelihood

A Bayesian approach to a problem is distinguished by the use of probability distributions over uncertain parameters. The resulting distributions over uncertain quantities represent a degree of belief in various possibilities [38]. Model fitting involves the fitting of parameters w/W to given model structure M_i or M_j such as the ones in equations 2-7 and 2-8.

$$y = \Phi(u)w + \Xi \qquad (M_i) \tag{2-7}$$

$$y = Net(u, W) + \Xi \qquad (M_j) \tag{2-8}$$

$$y = \quad \vdots$$

In equation 2-7, y is the real system's output, u the model input and Φ can be a set of basis functions or kernels. More relevant to this thesis, in model M_j, Net is the neural network function and W its connections' weights.

In an article on Bayesian interpolation by D.J.C. MacKay in 1992, two levels of inferences are presented [33]:

The **first level** of inference, a model M is chosen and represents a hypothesis H. At this level, the model is fitted to the data. In other words, we infer the parameter W from data D given the model structure M_i. Using Bayes' rule, this implies computing the posterior $P(W|D, M_i)$ as in equation 2-9 [33].

$$\overbrace{P(W|D, M_i)}^{\text{Posterior}} = \frac{\overbrace{P(D|W, M_i)}^{\text{Likelihood}}\overbrace{P(W|M_i)}^{\text{Prior}}}{\underbrace{\mathbf{P(D|M_i)}}_{\text{Evidence}}} \tag{2-9}$$

Note that the hypothesis does not only represent the model fitting choice but also can be the inference assumptions taken such as likelihood and prior forms.

The **second level** of inference consists of model comparison. This means choosing the most plausible model structure (hypothesis) chosen. Hence in Bayesian words, the posterior of the model M_i among models given data is evaluated and indicates this plausibility. The posterior of each model M_i in a space of models m is given using Bayes' rule in equation 2-10 [33].

$$P(M_i|D) = \frac{\mathbf{P(D|M_i)}P(M_i)}{P(D)} = \frac{\mathbf{P(D|M_i)}P(M_i)}{\sum_m P(D|M_k)P(M_k)} \tag{2-10}$$

The normalizing factor $P(D) = \sum_m P(D|M_k)P(M_k)$ is usually not included since the model space is typically not determined before hand. In addition, the prior to each model M_i is a

subjective prior and it is little common to assume unequal priors $P(M_i)$ on different models. Hence, using equation 2-10 it is easy to note that the posterior probability (the plausibility of the model) highly depends on the evidence [33].

The evidence of the model is also often referred to as a **marginal likelihood** it is the normalizing factor in equation 2-9 and can be expressed as:

$$\mathbf{P(D|M_i)} = \int P(D|W, M_i)P(W|M_i)dW \tag{2-11}$$

For the sake of the argument, instead of evaluating this integral, the posterior $P(W|D, M_i)$ is assumed to have a strong peak at W_{MP} and width ΔW and the priors are assumed uniform $P(W_{MP}|M_i) = \frac{1}{\Delta^0 W}$. This assumption is an adequate one, since given the data, the posterior has been demonstrated to sharpen up with respect to the prior [45, 65]. Equation 2-11 becomes:

$$\underbrace{\mathbf{P(D|M_i)}}_{\text{Evidence}} \approx \underbrace{\mathbf{P(D|W_{MP}, M_i)}}_{\text{Best Fit Data Likelihood}} \underbrace{\overbrace{P(W_{MP}|M_i)}^{\text{Priors on Weights}} \Delta W}_{\text{Occam's Factor}} = \mathbf{P(D|W_{MP}, M_i)} \underbrace{\frac{\Delta W}{\Delta^0 W}}_{\text{Occam's Factor}} \tag{2-12}$$

Figure 2-4 from MacKay article shows both the weights prior $P(W|M_i)$ and posterior $P(W|D, M_i)$ in equation 2-12.

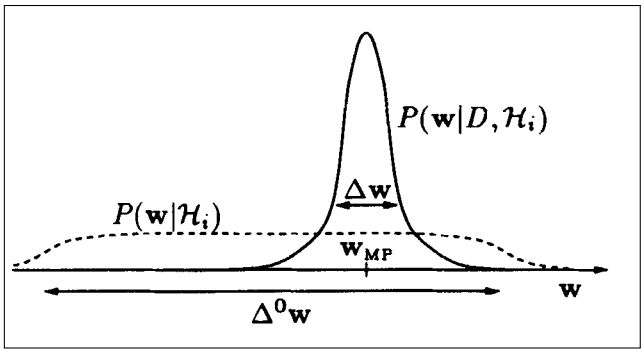


Figure 2-4: Approximations of the prior and posterior probability. Reprinted by permission from Springer Nature: Springer, Maximum Entropy and Bayesian Methods, Chapter 3 [33] by MacKay, David J. C. © 1992

Occam's factor $\frac{\Delta W}{\Delta^0 W}$ is at the basis of the **Occam's razor**: "Entities should not be multiplied without necessity" [56]. This razor essentially opposes the complexity associated with explaining phenomena.

The Occam's factor is less than one and hence penalizes the evidence. The more the number of parameters in model M_i, the more the priors that span a large range of $\Delta^0 W_k$ (k indicating the k^{th} parameter) and the more penalized the **marginal likelihood** becomes [33]. In addition, the Occam factor through the posterior width ΔW, makes the evidence favor coarser models. In other words, the evidence penalizes parameters that need to be fine-tuned.

But what about the likelihood? The best fit likelihood is the one that fits the data the best, no matter what prior information to the arrival of data is [47]. This means that having more

parameters can be rewarded by the likelihood function. More complex models, such as a parameter per data-point, may perform better in likelihood terms. In consequence, using it as a measure of inference can result in very complex models and data overfitting. With the marginal likelihood considered in the previous review of MacKay's work, a built in penalty for more complex models naturally exist with marginalization through the Occam's factor and balances out likelihood and complexity.

In other words, one can see the marginal likelihood increasing with the increase in model complexity caused by a data likelihood overwhelming the priors. However, at a certain level of complexity, the marginal likelihood (evidence of the model) starts decreasing because a thinner prior will be spread out across more and more parameters and their ranges [47].

2-2-2 Inference and Prediction

Starting from the general regression problem in equation 2-13 with design matrix Φ $(N \times M)$,

$$y = \Phi W + \Xi \tag{2-13}$$

a full Bayesian treatment consists of estimating the predictive distribution $\hat{y}$ given D, the data (previous observations) used for the inference of W and other parameters. This predictive distribution is computed in equation 2-14 [45] by marginalizing over the parameter W.

$$p(\hat{y}|D) = \int p(\hat{y}|W)p(W|D)dW \tag{2-14}$$

$p(\hat{y}|W)$ in equation 2-14 is the likelihood function defined also for $p(y|W)$ when computing the posterior $P(W|y)$ shown in equation 2-15 in analogy to equation 2-9.

$$P(W|D) = \frac{P(D|W)P(W)}{\mathbf{P}(\mathbf{D})} = \frac{P(D|W)P(W)}{\int P(D|W)P(W)dW} \tag{2-15}$$

Equation 2-15 is analogous to equation 2-9 for the specific identification problem in equation 2-13. For the sake of simplicity in writing, the conditionality on the model choice is dropped here.

The integral in equation 2-14 cannot be computed because, typically, the integral for the marginal likelihood in the denominator of equation 2-15 is intractable [47, 65] and/or the likelihood term is . There exists many approaches to compute or approximate the posterior of W and the posterior predictive distribution. These can be variational approximation, the choice of conjugate priors, Monte Carlo simulations and others [26, 66].

2-2-3 Likelihood and Sparsity Inducing Priors

To proceed with the inference procedure, one should choose the form of the likelihood function and the priors in equation 2-15. With the canonical form of the identification problem in equation 2-13 , arguably, the output noise Ξ can be assumed to be Gaussian $\mathcal{N}(0, \Pi)$. Hence, the likelihood function is very commonly chosen to be a Gaussian function with a covariance matrix Π [74, 47, 65, 71, 46].

$$p(y|W) = \mathcal{N}(y|\Phi W, \Pi) = \frac{1}{(2\pi)^{M/2}\Pi^{1/2}} \exp[-\frac{1}{2}(y - \Phi W)^T \Pi^{-1}(y - \Phi W)] \tag{2-16}$$

Note that the likelihood can be any function in the exponential family. Other likelihood functions can be Gamma, Dirichlet, Bernoulli, Poisson, Chi-squared and others [47].

The choice of the likelihood is not enough to render the integral in equation 2-15 tractable. The prior definition is crucial to the development of a Sparse Bayesian Learning algorithm. Typically, the likelihood rewards complexity, that is why one opts to introduce sparsity inducing priors. Prior forms that induce sparsity can be the Student's t-distribution or the Laplace distribution which places probability mass on the axial ridges of the parameter space. However these prior forms do not belong to exponential families and the posterior for these prior form remain intractable.

Another approach is the use of a relaxed variational approximation of the true prior $p(W)$. Non-stationary priors with a distinct variance ψ_n for each parameter W_n is used. This considered form of the prior yields a lower bound on the prior $p(W)$ under certain conditions [47, 45] discussed later in this section.

$$p(W) = \prod_{n=1}^{N} p(W_n) \geq \prod_{n=1}^{N} p(W_n, \psi_n) \tag{2-17}$$

$$p(W) = \max_{\psi_n > 0} \prod_{n=1}^{N} \mathcal{N}(W_n|0, \psi_n) p(\psi_n) = \max_{\psi > 0} \mathcal{N}(W|0, \Psi) p(\psi) \tag{2-18}$$

where $\psi = [\psi_1, \psi_2, .., \psi_n]$ and $\Psi = diag(\psi)$. $p(\psi)$ is called the hyperprior probability and is introduced because of the parameter ψ that needs to be inferred for a complete probabilistic specification of the model.

In other words, a relaxed prior can be formulated as follows:

$$p(W, \psi) = \mathcal{N}(W|0, \Psi) p(\psi) \leq p(W) \tag{2-19}$$

Conditions for the Gaussian relaxation of the prior:

This variational Gaussian relaxation of the priors is in fact only possible if the expression of the prior $p(W_n)$ is such that $\log(P(\sqrt{W_n}))$ is convex on $(0, \infty)$. This is in fact a property of the super-Gaussian distribution. In this case the hyperprior $p(\psi_n)$ takes a specific form [47, 44]. The main reason for that condition is in the derivation of the dual representation of the prior. If these conditions are met, the prior can be expressed as an upper envelope over its dual form leading to the variational approximation considered in equation 2-18. For more details over the proof, please refer to the article "Perspectives on Sparse Bayesian Learning" [45].

The representation of the prior with W and ψ does not only allow an approximate inference but is also known to induce sparsity. For a true prior in the form of a Student's t-distribution, the hyperprior is chosen to be the non-informative gamma distribution [74, 65, 4, 11].

$$p(\psi_n) = \texttt{Gamma}(\psi_\texttt{n}|\texttt{a}, \texttt{b}) = \frac{b^a \psi_n^{a-1}}{e^{b\psi_n} \mathbf{\Gamma}(a)} \qquad \texttt{a}, \texttt{b} \approx 0 \tag{2-20}$$

with $\mathbf{\Gamma}(\cdot)$ the gamma function.

This hierarchical nature of the prior "disguises its own character" [65]. To truly show the sparsity inducing property, the prior is integrated with respect to the hyperparameters ψ [65]:

$$p(W_n) = \int p(W_n|\psi_n)p(\psi_n)d\psi_n \tag{2-21}$$

$$= \frac{b^a \mathbf{\Gamma}(a+\frac{1}{2})}{\sqrt{2\pi}\mathbf{\Gamma}(a)}(b + \frac{W_n^2}{2})^{-(a+\frac{1}{2})} \tag{2-22}$$

This is the expression of a Student-t distribution, which has a sharp peak at 0. For non informative gamma prior ($a, b \approx 0$), the prior obtained is $p(W_n) \propto \frac{1}{|W_n|}$, hence the sparsity inducing property [65, 4].

2-2-4 Development of an Optimization Problem

For a fixed ψ, the conjugacy of the relaxed priors in equation 2-19 for the likelihood in equation 2-16 allows to obtain a relaxed posterior $P(W|y,\psi)$ in closed form. The resulting posterior mean and covariance are shown in equation 2-23 and 2-24 [47, 46]:

$$m_W = \Psi\Phi^T(\lambda I + \Phi\Psi\Phi^T)^{-1}y \tag{2-23}$$

$$\Sigma_W = \Psi - \Psi\Phi^T(\lambda I + \Phi\Psi\Phi^T)^{-1}\Phi \tag{2-24}$$

The issue however that remains is that of the choice of ψ. This choice affects how good the estimate of the relaxed posterior $P(W|y,\psi)$ is to the true posterior $p(W|y)$. With both W and ψ to be inferred, the full posterior is given by [47]:

$$\begin{aligned} p(W,\psi|y) &\propto p(W|y,\psi)p(\psi|y) \\ &= \mathcal{N}(m_W, \Sigma_W)p(\psi|y) \\ &= \mathcal{N}(m_W, \Sigma_W)\frac{p(y|\psi)p(\psi)}{p(y)} \propto \mathcal{N}(m_W, \Sigma_W)p(y|\psi)p(\psi) \end{aligned} \tag{2-25}$$

Note that $p(y)$ is dropped in equation 2-25 because of the independence from ψ.

It is important here to conclude that $p(y|\psi)p(\psi)$ refers to how good the inferred ψ explains the data. This quantity is in fact the evidence or the marginal likelihood and is at the basis of variational methods.

$$p(y|\psi)p(\psi) = \int p(y|W)p(W|\psi)p(\psi)dW = \int p(y|W)p(W,\psi)dW \tag{2-26}$$

Hence a good way to estimate ψ is to choose the one that minimizes the misaligned probability mass as follows:

$$\begin{aligned} \hat{\psi} &= \arg\min_{\psi\geq 0} \int p(y|W)|p(W) - p(W,\psi)|dW \\ &= \arg\max_{\psi\geq 0} \int p(y|W)p(W,\psi)dW \qquad p(W,\psi) \leq p(W) \\ &= \arg\max_{\psi\geq 0} \int \mathcal{N}(y|\Phi W, \Pi)\mathcal{N}(W|0,\Psi)p(\psi)dW \end{aligned} \tag{2-27}$$

Note that the noise covariance matrix Π *is assumed to be known in these inference steps and* Φ *is fixed.*

Figure 2-5, from Wipf article [71] , is a visualization of the search in equation 2-27.

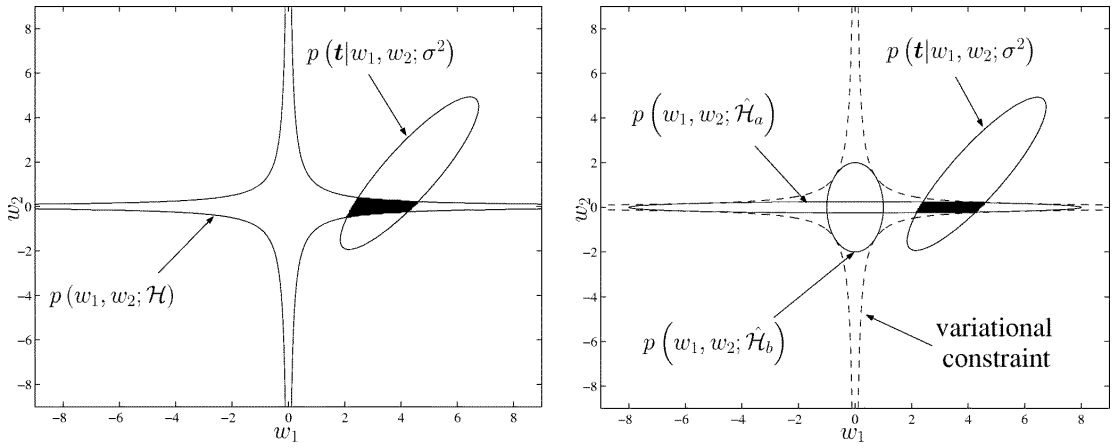


Figure 2-5: Comparison between full and approximate prior models equiprobability contours with the likelihood. From Wipf article "Perspectives on Sparse Bayesian Learning" [71] in IEEE Transactions on Signal Processing, vol. 52, no. 8, pp. 2160 © 2004 IEEE

Inspired by MacKay's framework described in section 2-2-1, the choice of ψ represents a space of hypothesis $\hat{H}$ that attempts to approximate the true prior hypothesis H. This space of hypotheses is in essence the space of possible ψ. The original inference problem consists of finding the aligning region of significant probability mass between the likelihood $p(t|W)$ and the true prior $p(W;H)$ (evidence maximization). This is shown in black fill in the left of figure 2-5. However, with the approximation of the prior, the search becomes a maximization of the evidence as in equation 2-27 among a space of hypotheses over the relaxed prior p(W;$\hat{H}$). In the right figure 2-5, the true prior hypothesis is plotted in dotted lines and is an upper bound over all variational approximation hypotheses (2 of them plotted in solid lines). It becomes easy to see that the hypothesis representing the narrow prior along the horizontal spine of p(W;H) is the better hypothesis. In other words, the hypothesis corresponding to the maximum evidence in 2-27 is $\hat{H}_a$.

In conclusion inferring ψ with equation 2-27, given data observations, allows to get a closed form on the relaxed posterior as in equations 2-23-2-24. The estimated parameter W can then be assigned to the posterior mean m_W. However, knowing the properties of the marginal likelihood discussed in section 2-2-1, another more general solution is to maximize the marginal likelihood in equation 2-27 jointly on W and ψ.

This maximization can be recast into an iterative reweighted l_1 or l_2 algorithm. Multiple treatments are very thoroughly described in Wei Pan's PHD Thesis: "Bayesian Learning for Nonlinear System Identification" [47]. These procedures inspire a big part of this thesis and will be discussed in later sections.

Finally, Sparse Bayesian Learning demonstrated its ability to generate sparse models capable of good generalization. Using a particular form of parameter prior, learning consists of the

marginal likelihood maximization with respect to the parameters. This is well known to be a *type II maximum likelihood estimation* and has been at the basis of relevance vector machines and Sparse Bayesian Learning [47, 65, 71, 4, 11].

2-2-5 Bayesian Deep Learning

Neural networks have long been criticized for their opaque representation of a process. Parameters in NNs often do not have a physical meaning hence, in statistical terms, they are non-parametric models [38]. These can also be over-parameterized, under-specified by the training data and can yield multiple different and highly performing models [70]. These disadvantages, however, seem to be well coupled with the advantages of Sparse Bayesian Learning. Through marginalization, the space of possible models can be more efficiently explored, sparsity can be enforced and uncertainties can be quantified.

Bayesian deep learning is often referred to as an attempt to demystify deep learning. Figure 2-6 shows how a perceptron can be interpreted in a probabilistic Bayesian framework on the right, as opposed to the conventional view on the left. With Bayesian deep learning, the connection weights are seen as probability distribution with a formulation of uncertainty, as opposed to the conventional view, where these consist of point estimates. This idea provides an opportunity to perform pruning or a form of dropout in the network based on not only magnitude, but also uncertainty [74, 47, 38].

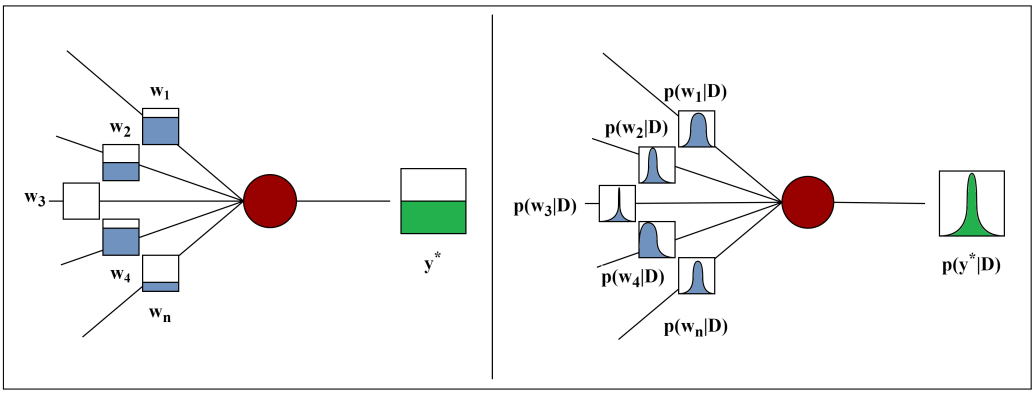


Figure 2-6: Figure showing the conventional and Bayesian views of a perceptron.

Uncertainties can also be quantified in the predictions made by the model through the posterior predictive distribution. In fact, Bayesian deep learning makes predictions in the form of a predictive distribution instead of point estimate. Neural networks tend to be data hungry systems and provide good approximations in training dataset regions, but can perform poorly in other regions where data lacks [70]. This uncertainty can be useful for instance in monitoring and decision making [17, 36].

Exact Bayesian inference is a computationally intractable problem, enforcing a trade-off in approximate methods between accuracy and computational tractability [62]. Multiple developments evolved to solve the intractable inference problem for neural networks. Among them are Laplace Approximation based inference [34], Hamiltonian Monte Carlo [38], Ensemble Learning [2]. For larger applications, more efficient approximate inference recently revived

this field, notably with Auto-Encoding Variational Bayes [25] and Dropout as a Bayesian Approximation [13].

Deep ensembles is a method recently developed [12] that succeeded in better representing the neural networks loss function landscape and the predictive uncertainties than some Bayesian methods [43]. The idea is to train a model using the maximum a posteriori estimate multiple times randomly initialized seeking different hypotheses corresponding to different local solutions obtained. However, the problem with MAP estimation, is that the prior is commonly chosen to be flat, rarely allowing any preferences in the parameter space and thus yielding imperfect solutions.

In this thesis, the Laplace approximation is adopted to solve the Bayesian intractable problem. This approach is known to be an efficient one, where the computational complexity slightly higher than MAP estimation and much less than other sampling methods [33]. However, this approximation is local and may not be sufficient to some applications [62]. That is why, inspired by deep ensembles, repeating the inference multiple times with random initialization would allow a better exploration of optimal models possible with a certain structure of the network.

The Laplace approximation requires the computation of an inverse hessian, which is infeasible for large networks. However, we stand motivated by the recent approximations provided by popular open-source software and the relation to hessian based network compression [28]. This approach has been successfully adopted by Zhou et al. [74] recent paper introducing a Bayesian approach to Neural Architecture Search. Inspired by the sparsity inducing priors [65], the paper formulates the inference problem as a iterative reweighted Lasso algorithm. Highly sparse Convolution Neural Network are generated with small search costs and competitive performance with state of the art algorithms on MNIST and CIFAR-10 Datasets.

Following similar steps as in sections 2-2-3, the main problem can be formulated as an iterative type II maximum likelihood procedure. This work intends to adopt and evaluate such treatment in the context of system identification.

Chapter 3

Manuscript

This Chapter includes a pre-print of a journal paper prepared for submission in Automatica.

Sparse Bayesian Deep Learning for Dynamic System Identification

Hongpeng Zhou [a] Chahine Ibrahim [b] Wei Pan [a]

[a] *Department of Cognitive Robotics, TU Delft*

[b] *Delft Center for Systems and Control, TU Delft*

Abstract

This paper proposes a complete Bayesian treatment of system identification using multi-layer perceptrons and Long Short Term Memory networks. Particularly, a practical iterative reweighed algorithm is derived and used to identify linear and non-linear processes. Artificial neural networks are powerful models known for excellent approximative ability and simple implementation, however, specific to system identification, these exhibit several challenges. These mappings are complex models, that are highly parameterized and that, reportedly, can overfit the data provided. Furthermore, inspired by the need to assist decision making, it proves beneficial to quantify uncertainty in model parameters and predictions. The Bayesian treatment addresses these challenges. It allows the quantification of uncertainties in estimated quantities and predictions, for which a practical method is derived. It also allows enforcement of structured sparsity by using group sparsity-inducing priors. The effectiveness of this treatment is successfully demonstrated on linear and non-linear real benchmark data of dynamic processes.

Key words: System Identification, Multi-Layer Perceptron, LSTM, Structured Sparsity, Bayesian Learning.

1 Introduction

System identification has long been an extensive subject of research in natural and social sciences. It is a mature field for both linear and non-linear systems, static and dynamic processes, with multiple approaches and nomenclatures. Among these, artificial neural networks (ANN) are prominent black-box models that are considerably studied in literature. Specific to dynamic non-linear system identification and control, a recent interest in these models has risen and multiple books have been published on the matter.

Such approach presents its advantages and disadvantages. In 1989, a paper on feed-forward neural networks mathematically proved the universal approximative capabilities of any measurable function, using one hidden layer, on a compact set [1]. Following it, multiple works found similar results for feed-forward neural networks [2] and recurrent neural networks in the context of dynamical systems [3]. This property makes it possible to approximate a process, without the need for basis function space selection. They are also simple and easy to implement. However, the exact model structure is hardly ever given. In fact, the choice of more than one layer for feed-forward neural networks might exhibit better convergence in some applications [4]. In addition to that, inevitable measurement noise and non-smooth characteristics of some non-linear processes all affect the model fit and its generalization property. Facing these challenges, Bayesian learning offers a perspective that tackles these issues, specifically the following: a) Over-fitting can be alleviated and model redundancies can be eliminated [5] by a choice of sparsity inducing prior distribution over parameters [6,7]; b) Model and prediction uncertainties can be quantified which is particularly useful in decision making and safety-critical applications such as autonomous driving [8] and structural health monitoring [9].

Diverse Bayesian system identification solutions have been developed the last decades. These differ in models considered (state-space, NARX, etc), the Bayesian treatment (Laplace approximation, Monte-Carlo sampling, etc) and applications. To name a few, Pan et al. (2016) proposed a practical sparse Bayesian approach to state-space identification of non-linear systems in the context of biochemical networks [5]. Jacobs et al. (2018) proposed a Bayesian identification of NARX models using variational inference with a demonstration on electroactive polymer [10]. Yuan et al. (2019) presented a framework for the identification of governing interactions and transition logics of subsystems in cyber-physical systems

by making use of Bayesian inference and pre-defined basis functions [11]. Chuiso et al. (2012) set out two approaches to system identification using Bayesian networks. The first combines kernel based stable splines and group Least Angle Regression and the other combines stable splines with the hyperprior definition in a fully Bayesian model [12]. However, specific to the use of Neural Networks as a model form, little attention was given to the identification of dynamic systems in a Bayesian framework.

In this paper, Multi-Layer Perceptron (MLP) and Long Short Term Memory networks (LSTM) are used as model forms. To model the network in a Bayesian perspective, group priors are introduced over network parameters to induce structured sparsity and the Laplace approximation is used to approximate the intractable integral of the evidence. The main contributions of this paper are:

- Derive of a practical iterative algorithm for system identification using Bayesian deep learning that can be used with both MLP and LSTM for linear and non-linear processes.
- Incorporate structured sparsity in the Bayesian formulation of the identification problem leading to compact sparse models.
- Test the effectiveness of this procedure on six real benchmarks datasets for linear/non-linear system identification by demonstrating close to state of the art simulation results, sparsity and uncertainty quantification.

The article is organised as follows. Section 2 formulates the identification problem using artificial neural networks and from a Bayesian perspective. All nomenclatures used can be found in this section. Section 3 describes the steps that leads to the proposed iterative procedure, defines structured sparsity and proposes a practical Monte-Carlo sampling method to estimate predictive uncertainty. Section 4 report the identification results on benchmark data of linear and non-linear processes. Finally, section 5 is a discussion of the results and section 6 concludes the paper.

2 Preliminaries

This section defines nomenclatures used in this paper, introduces the use of MLP, LSTM and the Bayesian approach to system identification. The chosen mathematical model structure is the map generated by training the network by $\texttt{Net}(\mathcal{W}, z)$, where $\mathcal{W}$ represents an array of all inferred connection weights in the network and z the input regressors of size $1 \times (l_y + l_u + 1)$. These are best defined by the prediction model in equation (1).

$$\hat{y}(t+1) = \texttt{Net}(\mathcal{W}, z(t+1)) + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \sigma^2) \quad (1)$$

The noise term ϵ is assumed normally distributed with mean zero and variance σ^2. The objective of this paper is the identification of dynamic systems. Hence the regressors used as inputs to these models will be defined as a combination of lagged elements of the systems inputs u and outputs y. The input lag is denoted l_u and output lag l_y, resulting in the expression $z(t+1) = [u(t-l_u), \cdots, u(t), u(t+1), y(t-l_y), \cdots, y(t-1), y(t)]^\top$.

The network is trained based on an one-step prediction approach, using stochastic gradient descent methods on a cost function to be derived later in section 3.

2.1 Multi-Layer Perceptron

A multi-layer perceptron (MLP) is a feed forward artificial neural network structure that usually consists of three classes of layers: an input layer $l = 1$, an output layer $l = L$ and intermediate hidden ones. The number of perceptrons in each layer is denoted as n_l. With the exception of the input and output nodes, every node is a perceptron with an activation function (e.g. sigmoid$(\cdot)$, relu$(\cdot)$, etc). These perceptrons are initially fully connected to the adjacent layers with affine maps. The MLP structure is visualized in Fig. 1. $\sigma(\cdot)$ refers to a user pre-

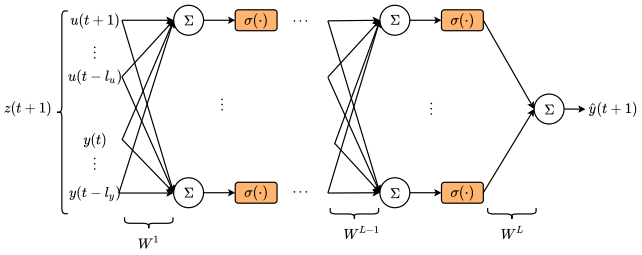


Fig. 1. Multi-Layer Perceptron with N layers

defined activation function. The $\mathcal{W}$ operator in equation (1), in the case of multi-layer perceptrons, is an vectorized form of all the network's connection weights $\mathcal{W} \triangleq [W^1_{11}, \cdots, W^1_{n_1 1}, \cdots, W^1_{n_1 n_2}, \cdots, W^L_{11}, \cdots, W^L_{n_{L-1} n_L}]$ of size $1 \times \kappa$.

By effect of the non-linear activation functions, the model is a non-linear map. For non-linear systems, relu networks are trained. In the case of linear systems, a linear activation function is chosen to generate a linear map.

2.2 Long Short Term Memory Networks

Long Short Term Memory networks (LSTM) is a class of recurrent neural networks. Unlike feed forward networks, this network includes feedback connections that allows arbitrary information from older inputs to linger longer. LSTM units have two memory states, the hidden

state and the cell state, representing respectively short-term and long-term memory information. The model's structure consist of three gates: input gate, output gate and forget gate. A sketch of a single layer LSTM network is given in Fig. 2. σ the sigmoid activation function and τ, the tanh activation function. The inputs to the first LSTM layer are the regressors z, instead of hidden states of the previous layer h^{l-1}, i.e. $h^0 = z$.

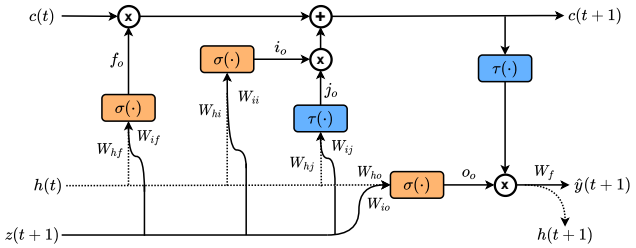


Fig. 2. Single layer Long Short Term Memory network.

In the case of LSTM network the $\mathcal{W}$ operator in equation (1) is an array in vector form of all connection weights in the network, including the affine operators of the hidden state. In other words, $\mathcal{W} \triangleq [\mathcal{W}_{ii}, \mathcal{W}_{ij}, \mathcal{W}_{io}, \mathcal{W}_{if}, \mathcal{W}_{hi}, \mathcal{W}_{hj}, \mathcal{W}_{ho}, \mathcal{W}_{hf}]$ of size $1 \times \kappa$.

Benefiting from the advantages of processing sequence of data and memorizing information, LSTM can also be used to solve the problem of nonlinear system identification [13]. Training LSTM networks is done using Back Propagation Through Time (BPTT), in which the network is unfolded in time and weights are updated based on an accumulation of gradients across time steps (check Appendix D).

2.3 Learning in a Bayesian Framework

Bayesian inference is statistical method that is essentially an extension of Baye's theorem. Specific to system identification using neural networks, the objective is to statistically infer the set of connection weights $\mathcal{W}$.
The Bayesian posterior estimation is given by Bayes rule in equation (2).

$$p(\mathcal{W}|\mathcal{D}, \mathcal{H}) = \frac{p(\mathcal{D}|\mathcal{W}, \mathcal{H})p(\mathcal{W}, \mathcal{H})}{p(\mathcal{D}|\mathcal{H})} \tag{2}$$

$p(y|\mathcal{W}, \mathcal{H})$ designates the likelihood function of the estimation data $\mathcal{D}$, $p(\mathcal{W}, \mathcal{H})$ the prior over the parameters to be inferred and $p(\mathcal{D}|\mathcal{H})$ is refered to as the evidence of the hypothesis $\mathcal{H}$ given $\mathcal{D}$. The hypothesis generally incorporates model and inference assumptions. For simplicity of notations, the hypothesis term is dropped in the rest of the paper.

The likelihood function is normal, centered around the network prediction with variance σ^2. The expression is given by equation (3). This corresponds to the Gaussian noise assumption taken on the variable y in equation (1).

$$\begin{aligned} p(\mathcal{D}|\mathcal{W}, \sigma^2) &= \prod_{t=1}^{T} \mathcal{N}(y(t)|\texttt{Net}(\mathcal{W}, z(t+1)), \sigma^2) \\ &= (2\pi\sigma^2)^{-\frac{T}{2}} \exp\big\{ -\mathbf{E}(\mathcal{W}, \sigma^2)\big\} \end{aligned} \tag{3}$$

where $\mathbf{E}(\mathcal{W}, \sigma^2)$ designates an energy loss function of the neural network in the form of a sum of squared errors (see equation (4)).

$$\mathbf{E}(\mathcal{W}, \sigma^2) = \frac{1}{2\sigma^2} \sum_{t=1}^{T} (y(t) - \text{Net}(\mathcal{W}, z(t+1)))^2 \tag{4}$$

A Gaussian relaxed variational approximation to the prior distribution is considered. The joint prior formulation constitutes a variational lower bound over the true prior $p(\mathcal{W})$ and is shown in equations (5)-(7) [14].

$$p(\mathcal{W}) \geq p(\mathcal{W}, \psi) \tag{5}$$

$$= \prod_{l=1}^{L} \prod_{a=1}^{n_{l-1}} \prod_{b=1}^{n_l} \mathcal{N}(W_{ab}^l|0, \psi_{ab}^l)\ \phi(\psi_{ab}^l) \tag{6}$$

$$= \mathcal{N}(\mathcal{W}|0, \Psi)\ \phi(\psi) \tag{7}$$

$\psi \triangleq [\psi_{11}^1, \cdots, \psi_{n_1 1}^1, \cdots, \psi_{n_1 n_2}^1, \cdots, \psi_{11}^L, \cdots, \psi_{n_{L-1} n_L}^L]$ and $\Psi \triangleq \text{diag}(\psi)$

Both $\mathcal{W}$ and ψ are parameters to be found. The inference can thus be seen on two levels. The prior distribution and the likelihood belong to exponential families, thus the prior is conjugate to the posterior for the defined likelihood. With this, the posterior over $\mathcal{W}$ can be analytically obtained in closed form. For Ψ, one can use the principle of minimizing the sum of misaligned probability mass [14] as follows:

$$\hat{\psi} = \underset{\psi \geq 0}{\text{argmin}} \int p(\mathcal{D}|\mathcal{W}, \sigma^2)|p(\mathcal{W}) - p(\mathcal{W}, \psi)|\ d\mathcal{W} \tag{8}$$

$$= \underset{\psi \geq 0}{\text{argmax}} \int p(\mathcal{D}|\mathcal{W}, \sigma^2) p(\mathcal{W}, \psi)\ d\mathcal{W} \tag{9}$$

Equation (8) becomes (9) by using the inequality in equation (5). The resulting problem is known as a type II maximum likelihood [7]. The maximization, however requires computing a well known intractable integral. There exists multiple approaches to this problem based on the Laplace approximation, expectation propagation and variational inference techniques. Among these, the Laplace approximation is adopted in this paper and the maximization in equation (9) is reformulated as a lasso iterative reweighted algorithm. Zhou et al. (2019) successfully adopted this Bayesian treatment (Laplace approximation) for neural architecture search in the con-

text of convolution neural networks. The models generated exhibited close to state of the art image classification performance with smaller search costs [15].

3 ALGORITHM DEVELOPMENT

3.1 Laplace Approximation

To compute the intractable integral in equation (9), the Laplace approximation is taken on the likelihood function (equation (3)). The energy function in equation (4) can be approximated by a second order Taylor series expansion around a set of connection weights $\mathcal{W}^*$ with operator $\Delta\mathcal{W} = \mathcal{W} - \mathcal{W}^*$.

$$\mathbf{E} \approx E(\mathcal{W}^*, \sigma^2) + \frac{1}{2}\Delta\mathcal{W}^T\mathbf{H}\Delta\mathcal{W} + \Delta\mathcal{W}^T\mathbf{g} \tag{10}$$

The resulting expression for the likelihood in a compact form is given by equations (11).

$$p(\mathcal{D}|\mathcal{W}, \sigma^2) = \mathbf{A}\big(\mathcal{W}^*, \sigma^2\big) \exp\Big\{ -\frac{1}{2}\mathcal{W}^T\mathbf{H}\mathcal{W} - \mathcal{W}^T\hat{\mathbf{g}}\Big\} \tag{11}$$
$$\hat{\mathbf{g}}\big(\mathcal{W}^*, \sigma^2\big) = \mathbf{g}\big(\mathcal{W}^*, \sigma^2\big) - \mathbf{H}\big(\mathcal{W}^*, \sigma^2\big)\mathcal{W}^*$$

where $\mathbf{H}\big(\mathcal{W}^*, \sigma^2\big)$ and $\mathbf{g}\big(\mathcal{W}^*, \sigma^2\big)$ are respectively the hessian and the gradient of the loss function E with respect to $\mathcal{W}$ at $\mathcal{W}^*$. Equation (11) is obtained by grouping elements independent of the target variable $\mathcal{W}$ in $\mathbf{A}\big(\mathcal{W}^*, \sigma^2\big)$. The approximated likelihood is an exponential of a quadratic function corresponding to the Taylor series expansion of the energy loss. This form can be recast into a Gaussian function. For a more detailed treatment of the Laplace approximation check appendix A.

In effect of the conjugacy of the prior and posterior for the likelihood, the posterior is Gaussian with mean and variance given by equation (12).

$$\mu_{\mathcal{W}} = \Sigma_{\mathcal{W}}\hat{\mathbf{g}} \qquad \Sigma_{\mathcal{W}} = \big[\mathbf{H} + \Psi^{-1}\big]^{-1} \tag{12}$$

3.2 Evidence Maximization

The evidence in equation (9) attempts to find the volume of the product $p(\mathcal{D}|\mathcal{W}, \sigma^2)p(\mathcal{W}, \psi)$, which is Gaussian and proportional to the posterior. Thus, one can approximate the evidence as the volume around the most probable value (here the posterior $\mu_{\mathcal{W}}$).

$$\hat{\psi} = \underset{\psi \geq 0}{\operatorname{argmax}} \int p(\mathcal{D}|\mathcal{W}, \sigma^2)p(\mathcal{W}|\psi)p(\psi)d\mathcal{W} \tag{13}$$
$$\approx \underset{\psi \geq 0}{\operatorname{argmax}} \underbrace{p(\mathcal{D}|\mu_{\mathcal{W}}, \sigma^2)}_{\text{Best Fit Likelihood}} \underbrace{p(\mu_{\mathcal{W}}|\psi)|\Sigma_{\mathcal{W}}|^{\frac{1}{2}}}_{\text{Occam Factor}} \tag{14}$$

In David Mackay's words, the evidence is approximated by the product of the data likelihood given the most probable weights, and the Occam factor [6]. It can also be interpreted as a Riemann approximation of the evidence, where the best fit likelihood represents the peak of the evidence and the Occam's factor is the Gaussian width/curvature around the peak [16].

The likelihood and the prior in equation (14) can take their respective expressions in equations (11) and (7). By realising that the posterior mean $\mu_{\mathcal{W}}$ maximizes $p(\mathcal{D}|\mathcal{W}, \sigma^2)p(\mathcal{W}|\psi)$, equation (14) can be rewritten into a joint maximization in $\mathcal{W}$ and ψ. By applying a $-2\log(\cdot)$ operation, the final joint objective function is formulated in Proposition 1.

Proposition 1: The evidence maximization in equation (9) can be recast into a joint minimization of an objective function $\mathcal{L}(\mathcal{W}, \psi, \sigma^2)$ given by:

$$\begin{aligned}\mathcal{L}(\mathcal{W}, \psi, \sigma^2) =& \mathcal{W}^T\mathbf{H}\mathcal{W} + 2\mathcal{W}^T\hat{\mathbf{g}} + \mathcal{W}^T\Psi^{-1}\mathcal{W} + \log|\Psi| \\ &+ \log|\mathbf{H} + \Psi^{-1}| - T\log(2\pi\sigma^2)\end{aligned} \tag{15}$$

For a more thorough mathematical derivation that leads to equation (15), please refer to appendix B.

3.3 Convex Concave Procedure

Proposition 2: The objective function in equation (15) can be seen as a sum of a convex u and concave v functions in ψ shown in equations (16)-(17).

$$u(\mathcal{W}, \psi) = \mathcal{W}^T\mathbf{H}\mathcal{W} + 2\mathcal{W}^T\hat{\mathbf{g}} + \mathcal{W}^T\Psi^{-1}\mathcal{W} \tag{16}$$
$$v(\psi) = \log|\Psi| + \log|\mathbf{H} + \Psi^{-1}| \tag{17}$$

This problem can be reformulated as a convex-concave procedure (CCCP) [17]. ψ is obtained with the minimization shown in equations (18)-(19).

$$\mathcal{W}(k+1) = \underset{\mathcal{W}}{\operatorname{argmin}}\, u\big(\mathcal{W}, \psi(k)\big) \tag{18}$$
$$\psi(k+1) = \underset{\psi \geq 0}{\operatorname{argmin}}\, u\big(\mathcal{W}(k+1), \psi\big) + \nabla_{\psi}v\big(\psi(k)\big)^T \cdot \psi \tag{19}$$

Proposition 3 : The analytical solution to equation (19) is given by the following iterative form:

$$\Sigma_{\mathcal{W}}(k) = \big(\mathbf{H}(k) + \Psi(k)^{-1}\big)^{-1} \tag{20}$$
$$\alpha^l_{ab}(k) = -\frac{\Sigma_{W^l_{ab}}(k)}{\psi^l_{ab}(k)^2} + \frac{1}{\psi^l_{ab}(k)} \tag{21}$$
$$\psi^l_{ab}(k+1) = \frac{|W^l_{ab}(k+1)|}{\omega^l_{ab}(k)} \tag{22}$$

Note that $\Sigma_{W^l_{ab}}(k)$ is the connection weight posterior variance, α the analytical expression for the gradient of

$v(\psi)$ in equation (19) and $\omega_{ab}^l \triangleq \sqrt{\alpha_{ab}^l}$. For the second part, finding $\mathcal{W}$ can be done by stochastic gradient descent on equation (18), which can be reformulated as a regularized neural network loss function.

$$\mathcal{W}(k+1) = \underset{\mathcal{W}}{\operatorname{argmin}}\, \mathcal{W}^T \mathbf{H} \mathcal{W} + 2\mathcal{W}^T \hat{\mathbf{g}} + \sum_{l=1}^{L} \sum_{a=1}^{n_{l-1}} \sum_{b=1}^{n_l} ||\omega_{ab}^l \cdot W_{ab}^l||_{l_1} \tag{23}$$

$$\approx \underset{W}{\operatorname{argmin}}\, \mathbf{E}(\cdot) + \lambda \sum_{l=1}^{L} \rho(\omega^l, W^l) \tag{24}$$

$\mathbf{E}(\cdot)$ designates the energy loss function defined in equation (4) and $\rho(\cdot)$ the regularization term. For more details over the convex analysis and Propositions 2 and 3 please refer to Appendix C.

3.4 Structured Sparsity and Regularization

The iterative procedure derived in section 3 includes an assumption on the independence and stationarity of connection weights (equation (6)) resulting in a shape-wise regularization as shown in Fig. 3(a). This drives individual connection weights to 0. In some applications, one may want to enforce more structured sparsity, by pre-defining groups and re-expressing the regularization term as a function of these groups [18]. This paper uses a structured regularization of rows and columns (Fig 3). The benefits of such approach, specific to this paper, is not only obtaining compact sparse models, but also, the suppression of input nodes in z that are deemed less pertinent without loss of accuracy.

To extend this approach to the Bayesian framework, one has to revisit the prior formulation. The prior of a weight matrix is formulated based on the designated group of weight matrices (row or column or both). These groups are considered independent but the connection weights of a specific group share the same prior Gaussian relaxation (see Fig. 3(b-d)). This results in a slightly different iterative update rule for the identification algorithm.

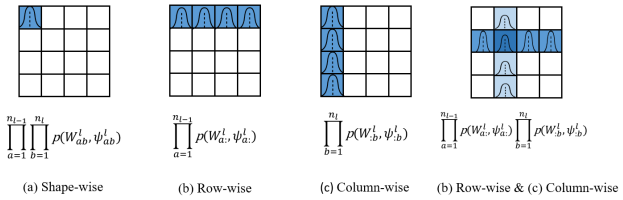


Fig. 3. Priors for structured sparsity of weight matrices.

For each of the cases shown in Fig. 3, the update rules for ψ, ω and the regularization function ρ are given in Table. 1. For more details on how the adopted group priors changes the regularization update rules to group Lasso regularizers please refer to Appendix F.

3.5 Identification Algorithm

A pseudo-code for the iterative procedure is given by Algorithm 1.

Algorithm 1 Sparse Bayesian Deep Learning Algorithm.

Input:
- Collect input-output data $u(t)$ and $y(t)$ for $t = 1, 2, \cdots, T$.
- Arrange input regressors according to chosen lags l_u, l_y.
- Define network structure (number of layers L, neurons per layer n_l and activations if it applies).
- Set regularization parameter λ (empirically tuned) and NN pruning thresholds κ_ψ, κ_W ($\approx 10^{-3}$).
- Set the number of repeated experiments M and identification cycles K_{max}.
- Initialize hyper-parameters $\Psi(0) = \mathbf{I}$ and $\omega(0) = \mathbb{1}$.

Output: Return the set of connection weights $\mathcal{W}$

for $m = 1$ to M **do**
  for $k = 1$ to $K_{\max}$ **do**
    (1) Stochastic Gradient Descent with loss function (ρ is defined in Table. 1):

$$\mathcal{W}(k+1) = \min_{\mathcal{W}} \mathbf{E}(\cdot) + \lambda \sum_{i=1}^{N} \boldsymbol{\rho}(\omega^l, W^l)$$

    (2) Update α according to equations (20)-(21)
    (3) Update ψ and ω according to Table. 1
    (4) Dynamic pruning:
    if $\psi_{ab}^l(k) < \kappa_\psi$ **or** $|W_{ab}^l(k)| < \kappa_W$ **then**
      prune $W_{ab}^l(k)$
    end if
  end for
  Simulate on validation data and choose model with smallest RMSE.
end for

Remark 1 *In the first identification iteration, regularization is conventional ($\omega(0) = \mathbb{1}$). That is, the first obtained model is a sparse model corresponding to conventional group lasso regularization method and sparser models result from next identification iterations.*

Remark 2 *The algorithm does not exhibit global convergence properties. In fact, it shares the local convergence properties (local minima, saddle point) of the adopted stochastic gradient descent method. This is because the Laplace approximation is a local approximation to the energy function $E(\cdot)$. In addition to that, pruning and the regularization techniques introduced are heuristics that help speed up the algorithm and improves convergence and optimality. Nonetheless, the identification experiments*

Table 1
Hyper-parameters update rule based on regularization technique.

Category	Prior Formulation	$\rho(\omega^l, W^l)$	ω^l	ψ^l
(a) Shape-wise	$\prod_{a=1}^{n_{l-1}} \prod_{b=1}^{n_l} p(W^l_{ab}, \psi^l_{ab})$	$\sum_{a=1}^{n_{l-1}} \sum_{b=1}^{n_l} \|\omega^l_{ab}(k) \cdot W^l_{ab}(k)\|_{l_1}$	$\omega^l_{ab}(k) = \sqrt{\alpha^l_{ab}(k)}$	$\psi^l_{ab}(k) = \frac{\|W^l_{ab}(k)\|}{\omega^l_{ab}(k-1)}$
(b) Row-wise	$\prod_{a=1}^{n_{l-1}} p(W^l_{a:}, \psi^l_{a:})$	$\sum_{a=1}^{n_{l-1}} \|\omega^l_{a:}(k) \cdot W^l_{a:}(k)\|_{l_2}$	$\omega^l_{a:}(k) = \sqrt{\sum_{b=1}^{n_l} \alpha^l_{ab}(k)}$	$\psi^l_{a:}(k) = \frac{\|W^l_{a:}(k)\|_2}{\omega^l_{a:}(k-1)}$
(c) Column-wise	$\prod_{b=1}^{n_i} p(W^l_{:b}, \psi^l_{:b})$	$\sum_{b=1}^{n_l} \|\omega^l_{:b}(k) \cdot W^l_{:b}(k)\|_{l_2}$	$\omega^l_{:b}(k) = \sqrt{\sum_{a=1}^{n_{l-1}} \alpha^l_{ab}(k)}$	$\psi^l_{:b}(k) = \frac{\|W^l_{:b}(k)\|_2}{\omega^l_{:b}(k-1)}$
(b) Row-wise + (c) Column-wise	$\prod_{a=1}^{n_{l-1}} p(W^l_{a:}, \psi^l_{a:}) \times \prod_{b=1}^{n_l} p(W^l_{:b}, \psi^l_{:b})$	$\sum_{a=1}^{n_{l-1}} \|\omega^l_{a:}(k) \cdot W^l_{a:}(k)\|_{l_2} + \sum_{b=1}^{n_l} \|\omega^l_{:b}(k) \cdot W^l_{:b}(k)\|_{l_2}$	$\omega^l_{a:}(k) = \sqrt{\sum_{b=1}^{n_l} \alpha^l_{ab}(k)}$ $\omega^l_{:b}(k) = \sqrt{\sum_{a=1}^{n_{l-1}} \alpha^l_{ab}(k)}$	$\psi^l_{a:}(k) = \frac{\|W^l_{a:}(k)\|_2}{\omega^l_{a:}(k-1)}$ $\psi^l_{:b}(k) = \frac{\|W^l_{:b}(k)\|_2}{\omega^l_{:b}(k-1)}$ $\psi^l_{ab}(k) = 1/(\frac{1}{\psi^l_{a:}(k)} + \frac{1}{\psi^l_{:b}(k)})$

are ran multiple times randomly initialized and the generated model with the best simulation validation performance is chosen.

Remark 3 *The Laplace approximation requires computing the Hessian of the loss function with respect to the networks parameters. This can be computationally heavy and infeasible for large networks. For fully-connected feed forward neural networks, an efficient Hessian computation method is proposed in [15]. For recurrent neural networks, an efficient Hessian computation method is proposed in appendix D.*

3.6 Making Predictions

In the Bayesian procedure, predictions are made using the posterior predictive distribution. It is given by equation (25).

$$p(\hat{y}|z, \mathcal{D}) = \int p(\hat{y}|\mathcal{W}, z)\ p(\mathcal{W}|\mathcal{D}) d\mathcal{W} \tag{25}$$

The first term of the integral is the likelihood of the unobserved value (prediction) conditional on the network prediction. The second term is the inferred posterior distribution over the weights $\mathcal{W}$ given the training data $\mathcal{D}$. Since the posterior predictive distribution is a convolution of Gaussian distributions, it is Gaussian. An unbiased estimate of the mean and variance can be obtained using Monte-Carlo integration methods [19,**?**]. A more thorough derivation using expected values is given in Appendix E. The resulting expressions are shown in equations (26)-(27).

$$\mu_{\hat{\mathbf{y}}} \approx \frac{1}{M} \sum_{m=1}^{M} \texttt{Net}(\mathcal{W}(m), z) \tag{26}$$

$$\mathbf{\Sigma}_{\hat{\mathbf{y}}} \approx \sigma^2 + \frac{1}{M} \sum_{m=1}^{M} \texttt{Net}(\mathcal{W}(m), z)^2 - \mu_{\hat{\mathbf{y}}}{}^T \mu_{\hat{\mathbf{y}}} \tag{27}$$

4 EXPERIMENTS

This section aims to summarize the identification experiments of three linear processes and three non-linear processes using the proposed algorithm. For linear systems, the identification procedure is repeated $M = 20$ times with $K_{max} = 6$ identification cycles. For non-linear systems, the identification is also repeated $M = 20$ times but with $K_{max} = 10$ identification cycles each. In Table 2, stands a summary of the model structure used for identification as well as the mean, standard deviation and minimum validation `RMSE` of the M best generated models, the percentage of sparse parameters in the best generated model and a reference to supplementary material. In the supplementary section, the benchmarks are described more thoroughly and the reader is supported with sparsity plots, simulation plots and plots of posterior predictive mean and uncertainty corresponding to the best generated model.

Three linear processes are identified, the Hairdryer, corresponding to the PT326 process trainer [20], a Heat exchanger [21] and Glass Tube manufacturing process [22]. The datasets of these processes are provided by Matlab in corresponding tutorials on linear system identification. The chosen best validated models are compared to methods used in the corresponding tutorials. Additional model structures used for the identification of the Hairdryer are taken from chapter 17.3 of [23] and run in Matlab. Check Table. 3 for the comparisons.

Three non-linear processes, the Cascaded Tanks [24], Coupled Electric Drives [25] and the Bouc-Wen hysteresis model [26] are also identified. Information and datasets of these benchmarks are compiled in the web page of the Workshop on Nonlinear System Identification Benchmarks.The models with the best validation performance are compared to best models obtained using conventional neural network methods for multiple experiments ($M = 20$) and previous works in literature for every benchmark in Tables. 3-4.

Table 2
Models trained to identify linear and non-linear processes with validation information

Process-Model	Layers-Units	Lags	$\texttt{RMSE}_{\texttt{val}}$ $(\mu \pm \sigma)$	$\texttt{RMSE}_{\texttt{val}}$ (min)	Sparsity	Supporting Material
Hairdryer-MLP	1 - 50	5	0.074 ± 0.0005	0.073	88.1%	Appendix G
Hairdryer-LSTM	1 - 10	5	0.093 ± 0.0166	0.081	93.5%	Appendix G
Heat Exchanger-MLP	1 - 50	150	0.086 ± 0.0002	0.086	99.3%	Appendix H
Heat Exchanger-LSTM	1 - 10	150	0.114 ± 0.0299	0.088	96.4%	Appendix H
GT Manufacturing-MLP	1 - 50	5	0.660 ± 0.0013	0.657	97.8%	Appendix I
GT Manufacturing-LSTM	1 - 10	5	0.671 ± 0.0019	0.669	99.0%	Appendix I
Cascaded Tanks-MLP	3 - 10	20	0.428 ± 0.1032	0.257	84.5%	Appendix J
Cascaded Tanks-LSTM	1 - 50	20	0.500 ± 0.1012	0.362	60.3%	Appendix J
CED-MLP	2 - 50	10	0.187 ± 0.0285 0.134 ± 0.0192	0.149 0.120	78.4%	Appendix K
CED-LSTM	1 - 10	10	0.155 ± 0.0257 0.126 ± 0.0201	0.121 0.097	72.8%	Appendix K
Bouc-Wen-MLP	2 - 50	10	0.171 ± 0.0087 0.133 ± 0.0315	0.148 0.117	38.5%	Appendix L
Bouc-Wen-LSTM	1 - 10	10	0.292 ± 0.0273 0.184 ± 0.0420	0.258 0.138	82.8%	Appendix L

5 DISCUSSION

In this section, the results will be discussed and analyzed in relation to the claims made on sparsity, uncertainty quantification and simulation results.

Sparsity: The obtained networks in all experiments are sparse models with, in most of the cases, a compact structured sparsity. According to Table. 2, sparsity was more prominent in the identified linear systems than in non-linear systems. This demonstrates that the non-linearity that the data exhibits, requires a higher complexity than in the linear case.

Starting with the linear systems, one can note that structured sparsity induced, in the case of the Heat Exchanger MLP and LSTM models, a recognised transport delay that characterizes this system. Figure 4 is an example of a sparsity plot of the Heat Exchanger identified LSTM Model. Furthermore, the LSTM models for the linear systems have complete operators pruned. This means that the cell state can well be regulated with fewer parameters than imposed from the initialized model structure in the case of the Heat Exchanger. Similar behavior is seen across linear benchmarks.

Structured sparsity was also observed in the generated networks for non-linear systems at the exception of the Bouc-Wen MLP identified model (Table. 2). One can note the sparser representation of the RNN network for the same benchmark. A possible reason for that is in the fact that the RNN networks are capable of representing the non measurable dynamic highly non-linear state

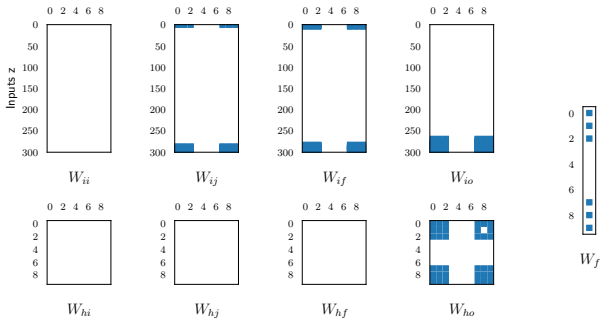


Fig. 4. Sparsity plot of the identified RNN for Heat Exchanger. (*Blue represents non-pruned connection weights*)

(hysteric force) with a sparser representation than the MLP network because of their temporal dynamic behavior. In addition to that, similarly to RNN models identified for linear systems, a lot of the parameters involving the hidden states are pruned. A possible explanation for this behaviour is in the fact that hidden states of LSTM units attempt to retain short-term information from the time-series that is also available as lagged elements in the input regressors.

Predictive Distributions: The posterior predictive distributions found for each of the models are a result of the forward propagation of the parameters' posterior uncertainty which, in turn, is obtained with the estimation data. Hence, if the validation data holds information that the model did not learn from estimation data, the posterior predictive distribution could spread

Table 3
Comparison of RMSE on identified linear systems with other works

Hairdryer	**RMSE [V]**
Transfer Function Estimation [20]	0.108
Subspace Identification [23]	0.105
ARMAX Model [23]	0.104
ARX Model [23]	0.103
LSTM without regularization	0.205
Bayesian LSTM	**0.081**
MLP without regularization	0.076
Bayesian MLP	**0.073**
Heat Exchanger	**RMSE** [$^{\circ}C$]
Transfer Function Estimation [21]	0.140
Process and Disturbance Model [21]	0.089
Process Model [21]	0.088
LSTM without regularization	0.158
Bayesian LSTM	**0.088**
MLP without regularization	0.092
Bayesian MLP	**0.086**
Glass Tube Manufacturing	**RMSE** [$\cdot$]
Subspace Identification [22]	0.688
ARX Model [22]	0.676
LSTM without regularization	1.056
Bayesian LSTM	**0.669**
MLP without regularization	0.663
Bayesian MLP	**0.657**

a bigger range of predictions.

In addition, in some cases, the generated models show an unevenly distributed predictive uncertainty related to non-linearities or disturbances characteristic of the process as well as regions where the model can be improved. Fig. 5 shows the identified model for Cascaded Tanks makes less robust predictions when overflow occurs. The Heat Exchanger shows evenly distributed predictions with uncertainty possibly coming from the ambient temperature disturbance. Furthermore, the model choice also affects the predictive distribution. Examples include the LSTM models identified for the Glass Tube Manufacturing Process and Cascaded Tanks. In these benchmarks, the identified MLP model provides robuster predictions than the identified RNN model.

Free Run Simulation Performance: The free run simulation is a good measure of the model's ability to represent a dynamic process by propagating a model's prediction error while forecasting. It is important to note that, for the studied linear processes, a non-regularized

Table 4
Comparison of RMSE on identified non-linear systems with other works

Cascaded Tanks	**RMSE [V]**	
LMN[a] with NFIR [27]	0.669	
Flexible State Space Model [28]	0.450	
Voltera Feedback Model [29]	0.397	
OEM[b] with NOMAD [30]	0.376	
Piecewise ARX Models [31]	0.350	
NLSS[c] [32]	0.343	
Tensor network B-splines [33]	0.302	
LSTM without regularization	0.494	
Bayesian LSTM	**0.362**	
MLP without regularization	0.432	
Bayesian MLP	**0.257**	
Coupled Electric Drives	**RMSE [ticks/s]**	
	Drive 1	Drive 2
Extended Fuzzy Logic [34]	0.150	0.092
Cascaded Splines [35]	0.216	0.110
TAG3P[d] [36]	-	0.128
RBFNN - FSDE[e] [37]	0.130	0.185
LSTM without regularization	0.149	0.131
Bayesian LSTM	**0.121**	**0.097**
MLP without regularization	0.206	0.111
Bayesian MLP	**0.149**	**0.120**
Bouc-Wen Hysteresis	**RMSE [mm]** ($\cdot 10^{-4}$)	
	Multisine	Sinesweep
LMN[a] with NFIR/NARX [27]	1.638	1.380
OEM with Nelder-Mead[b] [30]	0.468	0.019
Polynomial NLSS[c] [38]	0.187	0.120
TAG3P[d] [36]	-	0.652
Volterra Feedback Model [29]	0.875	0.639
LSTM without regularization	0.294	0.230
Bayesian LSTM	**0.258**	**0.138**
MLP without regularization	0.249	0.218
Bayesian MLP	**0.148**	**0.117**

[a] *Tree based Local Model Networks with external dynamics represented by NARX or NFIR.*
[b] *Output Error parametric Model estimation based on derivative free method.*
[c] *Non-Linear State Space model.*
[d] *Tree Adjoining Grammars*
[e] *Free Search Differential Evolution is used to determine the regressors.*

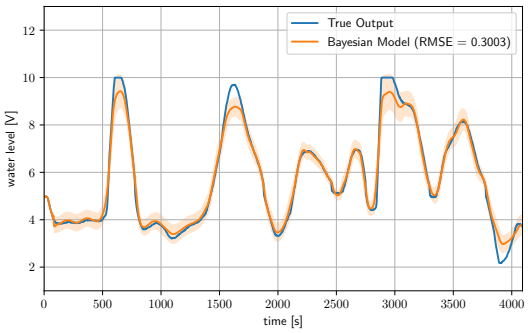


Fig. 5. Posterior predictive $\mu_y \pm 2 * \sigma_y$ of the identified Cascaded Tanks MLP.

LSTM performs poorly when compared to other identification methods. This supports previous concerns made on using LSTM for the identification of linear systems. In most presented applications, the Bayesian MLP model outperforms the Bayesian LSTM model with the exception of the Coupled Electric Drive.

Table. 2, shows the mean and the standard deviation of the validation simulation errors as well as the minimum corresponding to the best chosen model. The minimum is seen to fall close to the range of one standard deviation from the mean. In addition, the variance of validation errors for linear systems is overall less than non-linear systems and the variance of validation errors for MLP models is also less than LSTM models. A possible explanation is that the added complexity in the identification of non-linear processes and/or the usage of more complex non-linear structures (LSTM in this case), increases the likelihood of a convergence towards saddle points. This is mainly because the Laplace method adopted is a local approximation of the evidence, which is a limitation of the proposed method and justifies running the identification experiment M times.

Nonetheless, in every case (check Table. 3-4), the Bayesian approach to the identification of each of the benchmarks constitutes an improvement over the conventional MLP and LSTM methods in terms of simulation errors and pushes these methods to perform competitively against others in literature.

6 CONCLUSION

A Bayesian perspective to system identification has been discussed. An iterative procedure for dynamic system identification using Bayesian Neural Networks has been derived and evaluated with datasets of three linear and three non-linear dynamic processes. The Bayesian approach in this paper made use of the Laplace approximation to approximate the evidence, a formulation of group sparsity inducing priors to enforce sparsity and Monte-Carlo integration methods to estimate the posterior predictive distribution. The generated models for each of the dynamic processes are sparse models that performed competitively with other used system identification methods in a free run simulation setting. In addition to that, uncertainties in inferred predictions, through the posterior predictive distribution, and inferred connection weights, through their posterior distribution, were quantified. Future works include applications where these uncertainties can be used. In particular, these uncertainties can help redesign a data acquisition experiment to improve the model fit with another identification iteration. In addition, more prior knowledge of the physical model will be included in the design of the network structure, which when coupled with structured sparsity is expected to improve both sparsity and interpretability of the resulting models.

References

[1] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366, 1989.

[2] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

[3] Anton Maximilian Schäfer and Hans Georg Zimmermann. Recurrent neural networks are universal approximators. In *Proceedings of the 16th International Conference on Artificial Neural Networks - Volume Part I*, ICANN'06, page 632–640, Berlin, Heidelberg, 2006. Springer-Verlag.

[4] J. Sjöberg, H. Hjalmarsson, and L. Ljung. Neural networks in system identification. *IFAC Proceedings Volumes*, 27(8):359 – 382, 1994. IFAC Symposium on System Identification (SYSID'94), Copenhagen, Denmark, 4-6 July.

[5] Wei Pan, Ye Yuan, Jorge Gonçalves, and Guy-Bart Stan. A sparse bayesian approach to the identification of nonlinear state-space systems. *IEEE Transactions on Automatic Control*, 61(1):182–187, 2016.

[6] David J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

[7] Michael E. Tipping. *Bayesian Inference: An Introduction to Principles and Practice in Machine Learning*, pages 41–62. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[8] Rowan McAllister, Yarin Gal, Alex Kendall, Mark van der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4745–4753, 2017.

[9] Yong Huang, Changsong Shao, Biao Wu, James L. Beck, and Hui Li. State-of-the-art review on bayesian inference in structural system identification and damage assessment. *Advances in Structural Engineering*, 22(6):1329–1351, 2019.

[10] W. R. Jacobs, T. Baldacchino, T. Dodd, and S. R. Anderson. Sparse bayesian nonlinear system identification using variational inference. *IEEE Transactions on Automatic Control*, 63(12):4172–4187, 2018.

[11] Ye Yuan, Xiuchuan Tang, Wei Zhou, Wei Pan, Xiuting Li, Hai-Tao Zhang, Han Ding, and Jorge Goncalves. Data driven discovery of cyber physical systems. *Nature communications*, 10(1):1–9, 2019.

[12] Alessandro Chiuso and Gianluigi Pillonetto. A bayesian approach to sparse dynamic network identification. *Automatica*, 48(8):1553 – 1565, 2012.

[13] A Delgado, C Kambhampati, and Kevin Warwick. Dynamic recurrent neural network for system identification and control. *IEE Proceedings-Control Theory and Applications*, 142(4):307–314, 1995.

[14] Jason Palmer, Bhaskar D. Rao, and David P. Wipf. Perspectives on sparse bayesian learning. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 249–256. MIT Press, 2004.

[15] Hongpeng Zhou, Minghao Yang, Jun Wang, and Wei Pan. Bayesnas: A bayesian approach for neural architecture search, 2019.

[16] Ethan Goan and Clinton Fookes. *Bayesian Neural Networks: An Introduction and Survey*, pages 45–87. Springer International Publishing, Cham, 2020.

[17] A. L. Yuille and Anand Rangarajan. The concave-convex procedure (cccp). In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, page 1033–1040, Cambridge, MA, USA, 2001. MIT Press.

[18] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks, 2016.

[19] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[20] The MathWorks Incorporation. Estimating simple models from real laboratory process data. `https://nl.mathworks.com/help/ident/examples.html`. Accessed: 2020-11-25.

[21] The MathWorks Incorporation. Estimating transfer function models for a heat exchanger. `https://nl.mathworks.com/help/ident/examples.html`. Accessed: 2020-11-25.

[22] The MathWorks Incorporation. Glass tube manufacturing process. `https://nl.mathworks.com/help/ident/examples.html`. Accessed: 2020-11-25.

[23] Lennart Ljung. *System Identification (2nd Ed.): Theory for the User*. Prentice Hall PTR, USA, 1999.

[24] M. Schoukens, Per Mattsson, Torbjörn Wigren, and J.M.M.G. Noël. Cascaded tanks benchmark combining soft and hard nonlinearities. In *Workshop on Nonlinear System Identification Benchmarks : April 25-27, 2016, Brussels, Belgium*, pages 20–23, April 2016. 2016 Workshop on Nonlinear System Identification Benchmarks ; Conference date: 25-04-2016 Through 27-04-2016.

[25] T. Wigren and M. Schoukens. *Coupled electric drives data set and reference models*. Number 024 in Technical Report Uppsala Universitet. Uppsala University Sweden, November 2017.

[26] J.P. Noël and M. Schoukens. Hysteretic benchmark with a dynamic nonlinearity. In *Workshop on Nonlinear System Identification Benchmark: April 11-13, 2016, Liege, Belgium*, pages 7–14, April 2016. 2016 Workshop on Nonlinear System Identification Benchmarks ; Conference date: 25-04-2016 Through 27-04-2016.

[27] Julian Belz, Tobias Münker, Tim O. Heinz, Geritt Kampmann, and Oliver Nelles. Automatic modeling with local model networks for benchmark processes. *IFAC-PapersOnLine*, 50(1):470 – 475, 2017. 20th IFAC World Congress.

[28] Andreas Svensson and Thomas B. Schön. A flexible state–space model for learning nonlinear dynamical systems. *Automatica*, 80:189–199, 2017.

[29] Maarten Schoukens and Fritjof Griesing Scheiwe. Modeling nonlinear systems using a volterra feedback model. In *Workshop on Nonlinear System Identification Benchmarks*, 2016.

[30] Mathieu Brunot, Alexandre Janot, and Francisco Javier Carrillo. Continuous-time nonlinear systems identification with output error method based on derivative-free optimisation. In *IFAC World Congress 2017*, Toulouse, FR, July 2017.

[31] Per Mattsson, Dave Zachariah, and Petre Stoica. Identification of cascade water tanks using a pwarx model. *Mechanical Systems and Signal Processing*, 106:40 – 48, 2018.

[32] Rishi Relan, Koen Tiels, Anna Marconato, and Johan Schoukens. An unstructured flexible nonlinear model for the cascaded water-tanks benchmark. *IFAC-PapersOnLine*, 50(1):452–457, 2017. 20th IFAC World Congress.

[33] Ridvan Karagoz and Kim Batselier. Nonlinear system identification with regularized tensor network b-splines. *Automatica*, 122, 2020.

[34] F. Sabahi and M. R. Akbarzadeh-T. Extended fuzzy logic: Sets and systems. *IEEE Transactions on Fuzzy Systems*, 24(3):530–543, June 2016.

[35] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini. Novel cascade spline architectures for the identification of nonlinear systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(7):1825–1835, July 2015.

[36] Stefan-Cristian Nechita, Roland Toth, Dhruv Khandelwal, and Maarten Schoukens. Toolbox for discovering dynamic system relations via tag guided genetic programming, 2020.

[37] H. V. H. Ayala, L. F. da Cruz, R. Z. Freire, and L. dos Santos Coelho. Cascaded free search differential evolution applied to nonlinear system identification based on correlation functions and neural networks. In *2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, pages 1–7, Dec 2014.

[38] Alireza Fakhrizadeh Esfahani, Philippe Dreesen, Koen Tiels, Jean-Philippe Noël, and Johan Schoukens. Polynomial state-space model decoupling for the identification of hysteretic systems. *IFAC-PapersOnLine*, 50(1):458 – 463, 2017. 20th IFAC World Congress.

[39] Jorge Nocedal and Stephen J. Wright. *Trust-Region Methods*, pages 66–100. Springer New York, New York, NY, 2006.

[40] Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical gauss-newton optimisation for deep learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 557–565. JMLR.org, 2017.

[41] Radford M. Neal. *Introduction*, pages 1–28. Springer New York, New York, NY, 1996.

[42] V. Wertz, G. Bastin, and M. Haest. Identification of a glass tube drawing bench. *IFAC Proceedings Volumes*, 20(5, Part 10):333 – 338, 1987. 10th Triennial IFAC Congress on Automatic Control - 1987 Volume X, Munich, Germany, 27-31 July.

[43] Nicolò Vaiana, Salvatore Sessa, and Luciano Rosati. A generalized class of uniaxial rate-independent models for simulating asymmetric mechanical hysteresis phenomena. *Mechanical Systems and Signal Processing*, 146:106984, 2021.

[44] M. Schoukens and J.P. Noël. Three benchmarks addressing open challenges in nonlinear system identification. *IFAC-PapersOnLine*, 50(1):446 – 451, 2017. 20th IFAC World Congress.

A The Laplace Approximation

In this section, a more detailed mathematical description of the Laplace approximation adopted is given.

The likelihood is defined by a Gaussian function in section 3.1. The formulation is rewritten in equation (A.1):

$$p(\mathcal{D}|\mathcal{W},\sigma^2)=\prod_{t=1}^{T}\mathcal{N}(y(t)|\texttt{Net}(z(t),\mathcal{W}),\sigma^2)=(2\pi\sigma^2)^{-\frac{T}{2}}\exp\Big\{-\mathbf{E}(\mathcal{W},\sigma^2)\Big\} \tag{A.1}$$

$\mathbf{E}(\mathcal{W},\sigma^2)$ is denoted as the energy function, or the loss of the network given the data $\mathcal{D}$. It is given by:

$$\mathbf{E}(\mathcal{W},\sigma^2)=\frac{1}{2\sigma^2}\sum_{t=1}^{T}(y(t)-\text{Net}(z(t),\mathcal{W}))^2 \tag{A.2}$$

The expression $\texttt{Net}(\cdot)$ in equation (A.2), is the resulting network non-linear map. To compute the intractable integral for the evidence, the energy function can be expanded to a using a second order Taylor series expansion around $\mathcal{W}^*$.

$$\mathbf{E}(\mathcal{W},\sigma^2)\approx\mathbf{E}(\mathcal{W}^*,\sigma^2)+(\mathcal{W}-\mathcal{W}^*)^T\mathbf{g}(\mathcal{W}^*,\sigma^2)+\frac{1}{2}(\mathcal{W}-\mathcal{W}^*)^T\mathbf{H}(\mathcal{W}^*,\sigma^2)(\mathcal{W}-\mathcal{W}^*) \tag{A.3}$$

where $\mathbf{g}=\nabla\mathbf{E}(\mathcal{W},\sigma^2)|_{\mathcal{W}^*}$ and $\mathbf{H}=\nabla\nabla\mathbf{E}(\mathcal{W},\sigma^2)|_{\mathcal{W}^*}$. The quadratic expression is also adopted among Trust-Region methods, where a region is defined around the current iterate connection weights $\mathcal{W}$ and the the expansion in equation (A.3) is considered a reasonable local representation of the loss function [39]. With this expansion, the likelihood function becomes:

$$p(\mathcal{D}|\mathcal{W},\sigma^2)\approx(2\pi\sigma^2)^{-\frac{T}{2}}\exp\Big\{-\Big(\frac{1}{2}(\mathcal{W}-\mathcal{W}^*)^T\mathbf{H}(\mathcal{W}^*,\sigma^2)(\mathcal{W}-\mathcal{W}^*)+(\mathcal{W}-\mathcal{W}^*)^T\mathbf{g}(\mathcal{W}^*,\sigma^2)+\mathbf{E}(\mathcal{W}^*,\sigma^2)\Big)\Big\} \tag{A.4}$$

$$=(2\pi\sigma^2)^{-\frac{T}{2}}\exp\Big\{-\Big(\frac{1}{2}\mathcal{W}^T\mathbf{H}(\mathcal{W}^*,\sigma^2)\mathcal{W}+\mathcal{W}^T\big(\mathbf{g}(\mathcal{W}^*,\sigma^2)-\mathbf{H}(\mathcal{W}^*,\sigma^2)\mathcal{W}^*\big)\Big)\Big\}$$

$$\cdot\exp\Big\{-\Big(\frac{1}{2}\mathcal{W}^{*T}\mathbf{H}(\mathcal{W}^*,\sigma^2)\mathcal{W}^*-\mathcal{W}^{*T}\mathbf{g}(\mathcal{W}^*,\sigma^2)+\mathbf{E}(\mathcal{W}^*,\sigma^2)\Big)\Big\} \tag{A.5}$$

$$=\mathbf{A}(\mathcal{W}^*,\sigma^2)\cdot\exp\Big\{-\Big(\frac{1}{2}\mathcal{W}^T\mathbf{H}(\mathcal{W}^*,\sigma^2)\mathcal{W}+\mathcal{W}^T\hat{\mathbf{g}}(\mathcal{W}^*,\sigma^2)\Big)\Big\} \tag{A.6}$$

with,

$$\hat{\mathbf{g}}\big(\mathcal{W}^*,\sigma^2\big)=\mathbf{g}\big(\mathcal{W}^*,\sigma^2\big)-\mathbf{H}\big(\mathcal{W}^*,\sigma^2\big)\mathcal{W}^* \tag{A.7}$$

$$\mathbf{A}(\mathcal{W}^*,\sigma^2)=(2\pi\sigma^2)^{-\frac{T}{2}}\cdot\exp\Big\{-\Big(\frac{1}{2}\mathcal{W}^{*T}\mathbf{H}(\mathcal{W}^*,\sigma^2)\mathcal{W}^*-\mathcal{W}^{*T}\mathbf{g}(\mathcal{W}^*,\sigma^2)+\mathbf{E}(\mathcal{W}^*,\sigma^2)\Big)\Big\} \tag{A.8}$$

A Gaussian form can be easily recuperated from equation (A.6) by completing the square in the exponent. Before that, we define the following quantities:

$$\mathbf{B}(\mathcal{W}^*,\sigma^2)=\exp\Big\{\frac{1}{2}\hat{\mathbf{g}}(\mathcal{W}^*,\sigma^2)^T\mathbf{H}(\mathcal{W}^*,\sigma^2)\hat{\mathbf{g}}(\mathcal{W}^*,\sigma^2)\Big\} \tag{A.9}$$

$$\mathbf{C}(\mathcal{W}^*,\sigma^2)=(2\pi)^{\frac{n}{2}}|\mathbf{H}(\mathcal{W}^*,\sigma^2)|^{\frac{1}{2}} \tag{A.10}$$

$$p(\mathcal{D}|\mathcal{W},\sigma^2) \approx \mathbf{A}(\mathcal{W}^*,\sigma^2)\cdot\exp\Big\{-\Big(\frac{1}{2}\mathcal{W}^T\mathbf{H}(\mathcal{W}^*,\sigma^2)\mathcal{W}+\mathcal{W}^T\hat{\mathbf{g}}(\mathcal{W}^*,\sigma^2)\Big)\Big\}$$
$$\cdot\exp\Big\{\frac{1}{2}\hat{\mathbf{g}}(\mathcal{W}^*,\sigma^2)^T\mathbf{H}(\mathcal{W}^*,\sigma^2)\hat{\mathbf{g}}(\mathcal{W}^*,\sigma^2)-\frac{1}{2}\hat{\mathbf{g}}(\mathcal{W}^*,\sigma^2)^T\mathbf{H}(\mathcal{W}^*,\sigma^2)\hat{\mathbf{g}}(\mathcal{W}^*,\sigma^2)\Big\} \quad (A.11)$$
$$= \mathbf{A}(\mathcal{W}^*,\sigma^2)\cdot\mathbf{B}(\mathcal{W}^*,\sigma^2)\cdot\exp\Big\{-\Big(\frac{1}{2}\mathcal{W}^T\mathbf{H}(\mathcal{W}^*,\sigma^2)\mathcal{W}+\mathcal{W}^T\hat{\mathbf{g}}(\mathcal{W}^*,\sigma^2)$$
$$+\frac{1}{2}\hat{\mathbf{g}}(\mathcal{W}^*,\sigma^2)^T\mathbf{H}(\mathcal{W}^*,\sigma^2)\hat{\mathbf{g}}(\mathcal{W}^*,\sigma^2)\Big)\Big\} \quad (A.12)$$
$$= \mathbf{A}(\mathcal{W}^*,\sigma^2)\cdot\mathbf{B}(\mathcal{W}^*,\sigma^2)\cdot\mathbf{C}(\mathcal{W}^*,\sigma^2)\cdot\mathcal{N}(\mathcal{W}|\hat{\mathcal{W}},\mathbf{H}^{-1}(\mathcal{W}^*,\sigma^2)) \quad (A.13)$$

where $\hat{\mathcal{W}} = -\mathbf{H}^{-1}(\mathcal{W}^*,\sigma^2)\hat{\mathbf{g}}(\mathcal{W}^*,\sigma^2)$.

Given a Gaussian likelihood and a Gaussian prior defined in section 2.3, by effect of the conjugacy rule, the posterior is also Gaussian $\mathcal{N}(\mu_{\mathcal{W}},\Sigma_{\mathcal{W}})$.

$$\mu_{\mathcal{W}} = \big[\mathbf{H}(\mathcal{W}^*,\sigma^2)+\Psi^{-1}\big]^{-1}\hat{\mathbf{g}}(\mathcal{W}^*,\sigma^2) \qquad \Sigma_{\mathcal{W}} = \big[\mathbf{H}(\mathcal{W}^*,\sigma^2)+\Psi^{-1}\big]^{-1} \quad (A.14)$$

B Evidence Maximization

This section provides a mathematical proof of derived objective function. Starting from the maximization in equation (9), the likelihood and prior is replaced by their expressions in the preliminary section 2.3.

$$\int p(\mathcal{D}|\mathcal{W},\sigma^2)p(\mathcal{W}|\psi)p(\psi)\,d\mathcal{W} \quad (B.1)$$
$$= \int \mathbf{A}\cdot\exp\Big\{-\Big(\frac{1}{2}\mathcal{W}^T\mathbf{H}\mathcal{W}+\mathcal{W}^T\hat{\mathbf{g}}\Big)\Big\}\cdot\mathcal{N}(\mathcal{W}|0,\Psi)\cdot\phi(\psi)\,d\mathcal{W} \quad (B.2)$$
$$= \frac{\mathbf{A}}{(2\pi)^{K/2}|\Psi|^{\frac{1}{2}}}\cdot\int\exp\Big\{-\Big(\frac{1}{2}\mathcal{W}^T\mathbf{H}\mathcal{W}+\mathcal{W}^T\hat{\mathbf{g}}\Big)\Big\}\cdot\exp\Big\{-\Big(\frac{1}{2}\mathcal{W}^T\mathbf{\Psi^{-1}}\mathcal{W}\Big)\Big\}\,d\mathcal{W}\cdot\prod_{l=1}^{L}\prod_{a=1}^{n^{l-1}}\prod_{b=1}^{n^l}\phi(\psi_{ab}^l) \quad (B.3)$$
$$= \frac{\mathbf{A}}{(2\pi)^{K/2}|\Psi|^{\frac{1}{2}}}\cdot\int\exp\Big\{-\mathcal{E}(\mathcal{W},\sigma^2)\Big\}\,d\mathcal{W}\cdot\prod_{l=1}^{L}\prod_{a=1}^{n^{l-1}}\prod_{b=1}^{n^l}\phi(\psi_{ab}^l) \quad (B.4)$$

where,

$$\mathcal{E}(\mathcal{W},\sigma^2) = \frac{1}{2}\mathcal{W}^T\mathbf{H}(\mathcal{W}^*,\sigma^2)\mathcal{W}+\mathcal{W}^T\hat{\mathbf{g}}(\mathcal{W}^*,\sigma^2)+\frac{1}{2}\mathcal{W}^T\mathbf{\Psi^{-1}}\mathcal{W} \quad (B.5)$$

The integral in equation (B.1) is the integral of the product $p(\mathcal{D}|\mathcal{W},\sigma^2)p(\mathcal{W}|\psi)$, which is proportional to the posterior $p(\mathcal{W}|\mathcal{D},\psi)$. In most applications, the posterior peaks with respect to the prior, and the evidence can be approximated by the posterior volume. This approximation is analogous to the usage of the Laplace approximation of the posterior in David MacKay's Bayesian framework [6].

$$\int p(\mathcal{D}|\mathcal{W},\sigma^2)p(\mathcal{W}|\psi)\,d\mathcal{W} \approx p(\mathcal{D}|\mu_{\mathcal{W}},\sigma^2)p(\mu_{\mathcal{W}}|\psi)\cdot|\Sigma_{\mathcal{W}}|^{\frac{1}{2}}\cdot(2\pi)^{\kappa/2} \quad (B.6)$$
$$\iff \int\exp\Big\{-\mathcal{E}(\mathcal{W},\sigma^2)\Big\}\,d\mathcal{W} \approx \exp\Big\{-\mathcal{E}(\mu_{\mathcal{W}},\sigma^2)\Big\}\cdot|\Sigma_{\mathcal{W}}|^{\frac{1}{2}}\cdot(2\pi)^{\kappa/2} \quad (B.7)$$

where,

$$\mathcal{E}(\mu_{\mathcal{W}}, \sigma^2) = \frac{1}{2}\mu_{\mathcal{W}}^T \mathbf{H}(\mathcal{W}^*, \sigma^2)\mu_{\mathcal{W}} + \mu_{\mathcal{W}}^T \hat{\mathbf{g}}(\mathcal{W}^*, \sigma^2) + \frac{1}{2}\mu_{\mathcal{W}}^T \mathbf{\Psi^{-1}} \mu_{\mathcal{W}} \tag{B.8}$$

$$= \min_{\mathcal{W}} \frac{1}{2}\mathcal{W}^T \mathbf{H}(\mathcal{W}^*, \sigma^2)\mathcal{W} + \mathcal{W}^T \hat{\mathbf{g}}(\mathcal{W}^*, \sigma^2) + \frac{1}{2}\mathcal{W}^T \mathbf{\Psi^{-1}} \mathcal{W} \tag{B.9}$$

Hence the maximization of the evidence becomes the maximization in equation (B.10) below.

$$\psi = \underset{\psi>0}{\operatorname{argmax}} \frac{\mathbf{A}(\mathcal{W}^*, \sigma^2)}{(2\pi)^{\kappa/2}|\Psi|^{\frac{1}{2}}} \cdot \exp\big\{-\mathcal{E}(\mu_{\mathcal{W}}, \sigma^2)\big\} \cdot |\Sigma_{\mathcal{W}}|^{\frac{1}{2}} \cdot \prod_{l=1}^{L}\prod_{a=1}^{n^{l-1}}\prod_{b=1}^{n^l} \phi(\psi_{ab}^l) \tag{B.10}$$

By applying a $-2\log(\cdot)$ operation and using equation (B.9), one obtains

$$\psi = \underset{\psi>0}{\operatorname{argmin}} -2\log\left[\frac{\mathbf{A}(\mathcal{W}^*, \sigma^2)}{(2\pi)^{\kappa/2}|\Psi|^{\frac{1}{2}}} \cdot \exp\big\{-\mathcal{E}(\mu_{\mathcal{W}}, \sigma^2)\big\} \cdot |\Sigma_{\mathcal{W}}|^{\frac{1}{2}} \cdot \prod_{l=1}^{L}\prod_{a=1}^{n^{l-1}}\prod_{b=1}^{n^l} \phi(\psi_{ab}^l)\right] \tag{B.11}$$

$$= \underset{\psi>0}{\operatorname{argmin}} -2\log(\mathbf{A}(\mathcal{W}^*, \sigma^2)) + \mathcal{E}(\mu_{\mathcal{W}}, \sigma^2) + \log|\Psi| - \log|\Sigma_{\mathcal{W}}| - 2\sum_{l=1}^{L}\sum_{a=1}^{n^{l-1}}\sum_{b=1}^{n^l}\log(\phi(\psi_{ab}^l)) \tag{B.12}$$

$$\mathcal{W}, \psi = \underset{\mathcal{W},\psi>0}{\operatorname{argmin}} \frac{1}{2}\mathcal{W}^T \mathbf{H}(\mathcal{W}^*, \sigma^2)\mathcal{W} + \mathcal{W}^T \hat{\mathbf{g}}(\mathcal{W}^*, \sigma^2) + \frac{1}{2}\mathcal{W}^T \mathbf{\Psi^{-1}} \mathcal{W} + \log|\Psi| + \log|\mathbf{H}(\mathcal{W}^*, \sigma^2) + \Psi^{-1}| \tag{B.13}$$

$$- 2\log(\mathbf{A}(\mathcal{W}^*, \sigma^2)) - 2\sum_{l=1}^{L}\sum_{a=1}^{n^{l-1}}\sum_{b=1}^{n^l}\log(\phi(\psi_{ab}^l))$$

Since the hyperprior $\phi(\psi)$ is a non-informative hyper-prior, the final objective function is given by:

$$\mathcal{L}(\mathcal{W}, \psi, \sigma^2) = \mathcal{W}^T \mathbf{H}\mathcal{W} + 2\mathcal{W}^T \hat{\mathbf{g}} + \mathcal{W}^T \Psi^{-1}\mathcal{W} + \log|\Psi| + \log|\mathbf{H} + \Psi^{-1}| - T\log(2\pi\sigma^2) \tag{B.14}$$

C Convex Analysis and Iterative Solution

This section intends to proof propositions 2 and 3, that help reformulate the optimization in equation (15) into a convex one. The objective function in equation (B.14) is a sum of two functions u and v, given by:

$$u(\mathcal{W}, \psi) = \mathcal{W}^T \mathbf{H}(\mathcal{W}^*, \sigma^2)\mathcal{W} + 2\mathcal{W}^T \hat{\mathbf{g}}(\mathcal{W}^*, \sigma^2) + \mathcal{W}^T \Psi^{-1}\mathcal{W} \tag{C.1}$$

$$v(\psi) = \log|\Psi| + \log|\mathbf{H}(\mathcal{W}^*, \sigma^2) + \Psi^{-1}| \tag{C.2}$$

$\mathcal{W}^T\Psi^{-1}\mathcal{W}$ is positive definite, since $\psi > 0$, thus u is convex in Ψ. v can be reformulated as a log-determinant of an affine function of Ψ. By using the Schur complement determinant identities,

$$|\Psi||\mathbf{H}(\mathcal{W}^*, \sigma^2) + \Psi^{-1}| = \begin{vmatrix} \mathbf{H}(\mathcal{W}^*, \sigma^2) & \\ & -\Psi \end{vmatrix} = |\mathbf{H}(\mathcal{W}^*, \sigma^2)||\mathbf{H}^{-1}(\mathcal{W}^*, \sigma^2) + \Psi| \tag{C.3}$$

and taking the log of equation (C.3),

$$\log|\Psi| + \log|\mathbf{H}(\mathcal{W}^*, \sigma^2) + \Psi^{-1}| = \log|\mathbf{H}(\mathcal{W}^*, \sigma^2)| + \log|\mathbf{H}^{-1}(\mathcal{W}^*, \sigma^2) + \Psi| \tag{C.4}$$

one finds an equivalent expression of v that is concave in Ψ (equation (C.4)) leading to proposition 2.

Given these properties of the objective function, the optimization can be re-expressed iteratively as a convex-concave procedure (CCCP) [17] formulated in equations (18)-(19). These become:

$$\mathcal{W}(k+1) = \underset{\mathcal{W}}{\operatorname{argmin}}\, \mathcal{W}^T \mathbf{H}(\mathcal{W}^*, \sigma^2)\mathcal{W} + 2\mathcal{W}^T \hat{\mathbf{g}}(\mathcal{W}^*, \sigma^2) + \mathcal{W}^T \Psi^{-1}(k)\mathcal{W} \tag{C.5}$$

$$\psi(k+1) = \underset{\psi \geq 0}{\operatorname{argmin}}\, \mathcal{W}^T(k+1)\Psi^{-1}\mathcal{W}(k+1) + \alpha(k) \cdot \psi \tag{C.6}$$

where $\alpha(k) = \nabla_\psi v\big(\psi(k)\big)^T$ is the gradient of v evaluated at the current iterate $\psi(k)$. Using the chain rule, its analytic form is given by:

$$\alpha(k) = \nabla_\psi \Big(\log|\Psi| + \log|\mathbf{H}(\mathcal{W}^*, \sigma^2) + \Psi^{-1}| \Big)\Big|_{\psi=\psi(k)} \tag{C.7}$$

$$= -\operatorname{diag}\Big(\Psi^{-1}(k)\Big) \circ \operatorname{diag}\left(\Big(\mathbf{H}(\mathcal{W}^*, \sigma^2) + \Psi^{-1}(k)\Big)^{-1}\right) \circ \operatorname{diag}\Big(\Psi^{-1}(k)\Big) + \operatorname{diag}\Big(\Psi^{-1}(k)\Big) \tag{C.8}$$

$$= \Big[\alpha^1_{11}, \cdots, \alpha^1_{n_1 1}, \cdots, \alpha^1_{n_1 n_2}, \cdots, \alpha^L_{11}, \cdots, \alpha^L_{n_{L-1}1}, \cdots, \alpha^L_{n_{L-1}n_L}\Big] \tag{C.9}$$

$\circ$ is the point-wise Hadamard product. Since Ψ is a diagonal matrix, equation (C.6) can be expressed per connection independently. For that, first, the following expressions are given.

$$\Sigma_{\mathcal{W}}(k) = \Big(\mathbf{H}(\mathcal{W}^*, \sigma^2) + \Psi^{-1}\Big)^{-1} \tag{C.10}$$

$$\alpha^l_{ab}(k) = -\frac{\Sigma_{\mathcal{W}^l_{ab}}(k)}{\psi^l_{ab}(k)^2} + \frac{1}{\psi^l_{ab}(k)} \tag{C.11}$$

The optimization in equation (C.6) becomes

$$\psi^l_{ab}(k+1) = \underset{\psi \geq 0}{\operatorname{argmin}}\, \frac{W^l_{ab}(k+1)^2}{\psi} + \alpha^l_{ab}(k) \cdot \psi \tag{C.12}$$

$$\psi^l_{ab}(k+1) = \underset{\psi \geq 0}{\operatorname{argmin}}\, \frac{W^l_{ab}(k+1)^2}{\psi} - 2\Big|\sqrt{\alpha^l_{ab}(k)} \cdot W^l_{ab}(k+1)\Big| + \alpha^l_{ab}(k) \cdot \psi \tag{C.13}$$

$$\psi^l_{ab}(k+1) = \underset{\psi \geq 0}{\operatorname{argmin}}\, \Big(|W^l_{ab}(k+1)| - \sqrt{\alpha^l_{ab}(k)} \cdot \psi\Big)^2 \tag{C.14}$$

and the analytical solution is thus $\psi^l_{ab}(k+1) = \frac{|W^l_{ab}(k+1)|}{\omega^l_{ab}(k)}$ where $\omega^l_{ab}(k) = \sqrt{\alpha^l_{ab}(k)}$.

Finally, by plugging this solution into equation (C.5), the second part of the CCCP can be reformulated as an l_1 regularized cost function of the network.

$$\mathcal{W}(k+1) = \underset{\mathcal{W}}{\operatorname{argmin}}\, \mathcal{W}^T \mathbf{H}(\mathcal{W}^*, \sigma^2)\mathcal{W} + 2\mathcal{W}^T \hat{\mathbf{g}}(\mathcal{W}^*, \sigma^2) + \sum_{l=1}^{L} \sum_{a=1}^{n_{l-1}} \sum_{b=1}^{n_l} ||\omega^l_{ab}(k) \cdot W^l_{ab}||_{l_1} \tag{C.15}$$

$$= \underset{\mathcal{W}}{\operatorname{argmin}}\, \mathbf{E}(\mathcal{W}^*, \sigma^2) + (\mathcal{W} - \mathcal{W}^*)^T \mathbf{g}(\mathcal{W}^*, \sigma^2) + \frac{1}{2}(\mathcal{W} - \mathcal{W}^*)^T \mathbf{H}(\mathcal{W}^*, \sigma^2) \tag{C.16}$$

$$+ 2\sum_{l=1}^{L} \sum_{a=1}^{n_{l-1}} \sum_{b=1}^{n_l} ||\omega^l_{ab}(k) \cdot W^l_{ab}||_{l_1}$$

$$\approx \underset{\mathcal{W}}{\operatorname{argmin}}\, \mathbf{E}(\mathcal{W}, \sigma^2) + \lambda \sum_{l=1}^{L} \mathbf{R}(\omega^l, W^l) \tag{C.17}$$

D Hessian Computation for a Recurrent Layer

In this work, as we adopt Laplace approximation method to calculate the posterior distribution, the Hessian of weight matrices within the neural network should be obtained. Previous work [40] and [15] have proposed the efficient recursive method to compute Hessian for a Fully-connected layer and convolutional layer, respectively. Inspired by these two works, we propose a recursive and efficient method to compute the Hessian of a recurrent layer.

D.1 Backward propagation through time (BPTT) process

As we know, a LSTM cell is a special form of the recurrent neural network and is equivalent to a FC neural network by unfolding itself over the time sequence. For convenience of explanation, we use a simplified RNN structure to illustrate the Hessian calculation process. As shown in Figure D.1, we denote $z(t)$, $h(t)$ and $y(t)$ as the input, hidden state and output of the time step t, respectively. The behaviour of this RNN layer can be described by:

$$h(t) = \sigma(W_i z(t) + W_h h(t-1)) \tag{D.1}$$

$$y(t) = W_o h(t) \tag{D.2}$$

where W_i, W_h and W_o represent the weight matrix of the input layer, hidden layer and output layer. σ is the activation function.

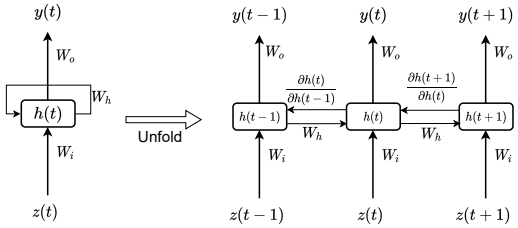


Fig. D.1. An unrolled RNN layer.

The hessian should be calculated through a backward propagation through time (BPTT) process. For the gradient update, if we use W to represent any weight matrix within a RNN, then its gradient is:

$$\frac{\partial \mathcal{L}}{\partial W} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}(t)}{\partial W} = \sum_{t=1}^{T} \sum_{k=1}^{t} \left(\frac{\partial \mathcal{L}(t)}{\partial h(t)} \frac{\partial h(t)}{\partial h(k)} \frac{\partial h(k)}{\partial W}\right) \tag{D.3}$$

where $L(t)$ is the loss at time t which is calculated as (24). To realize the backward propagation of gradient information from time step t to time step k, we have

$$\frac{\partial h(t)}{\partial h(k)} = \prod_{i=k+1}^{t} \frac{\partial h(i)}{\partial h(i-1)} \tag{D.4}$$

D.2 Hessian update process

Refer to [40] and the BPTT process, the Hessian for W_o is:

$$\mathbf{H}_o = \frac{1}{T} \sum_{t=1}^{T} \mathbf{H}_o^t \tag{D.5}$$

where Hessian $\mathbf{H}_o^t = (h(t))^2 \otimes H_o^t$ and H_o^t is the pre-activation Hessian of W_o.

Suppose τ is the backward propagation time horizon, the Hessian of W_h is computed as:

$$\mathbf{H}_h = \frac{1}{T \times \tau} \sum_{t=1}^{T} \sum_{j=t-\tau+1}^{t} \mathbf{H}_h^{t,j} \tag{D.6}$$

$$\mathbf{H}_h^{t,j} = (z(t-1))^2 \otimes H_h^{t,j} \tag{D.7}$$

where $\mathbf{H}_h^{t,j}$ and $H_h^{t,j}$ represent the Hessian and the pre-activation Hessian with $H_x^{t,j}$:

$$H_h^{t,j} = B^2 \circ (((W_h)^\top)^2 H_h^{t,j+1}) + D \tag{D.8}$$

where

$$B = \sigma'(h(t)), \quad D = \sigma''(h(t)) \circ \frac{\partial L}{\partial h(t)} \tag{D.9}$$

As the pre-activation Hessian H_h and Hessian $\mathbf{H}_h$ will be imparted along with the backward propagation process over the time horizon τ, (D.6) can be rewrote as:

$$\mathbf{H}_h = \frac{1}{T \times \tau} \sum_{t=1}^{T} \sum_{k=t-\tau+1}^{t} \mathbf{H}_h^{t,k} \tag{D.10}$$

Similarly, the Hessian of the W_i could be computed as:

$$\mathbf{H}_i = \frac{1}{T \times \tau} \sum_{t=1}^{T} \sum_{k=t-\tau+1}^{t} \mathbf{H}_i^{t,k} \tag{D.11}$$

where $\mathbf{H}_i^{t,k} = (z(k-1))^2 \otimes H_i^{t,k}$. $H_i^{t,k}$ can be computed as:

$$H_i^{t,k} = \prod_{j=k+1}^{t} B^2 \circ (((W_i)^\top)^2 H_i^{j,j-1}) \tag{D.12}$$

$$B = \sigma'(h(j)), \quad D = \sigma''(h(j)) \circ \frac{\partial L}{\partial h(j)} \tag{D.13}$$

E Posterior Predictive Mean and Variance

To find the expected value of the prediction, the expression of the posterior predictive distribution in equation (25) is used, and given that the likelihood is defined as a normal distribution one obtains:

$$\mathbb{E}\big[\hat{y}\big] = \int \hat{y}\ p(\hat{y}|z, \mathcal{D}) d\hat{y} \tag{E.1}$$

$$= \int \Big(\int \hat{y}\ p(\hat{y}|\mathcal{W}, z) d\hat{y} \Big)\ p(\mathcal{W}|\mathcal{D}) d\mathcal{W} \tag{E.2}$$

$$= \int \texttt{Net}(\mathcal{W}, z)\ p(\mathcal{W}|\mathcal{D}) d\mathcal{W} \tag{E.3}$$

Using the inferred posterior distribution over the weights, one can approximate this integral by Monte-Carlo sampling methods [19,41]. An unbiased estimate of the prediction is given by the average predictions using $\mathcal{W}$ sampled by the posterior M times.

$$\mu_{\hat{\mathbf{y}}} \approx \frac{1}{M} \sum_{m=1}^{M} \texttt{Net}(\mathcal{W}(m), z) \tag{E.4}$$

In an analogous way, to estimate the variance in the posterior predictive distribution, the expected value $\mathbb{E}\big[\hat{y}^T\hat{y}\big]$ is analytically derived as follows in equations (E.5)-(E.7).

$$\mathbb{E}\big[\hat{y}^T\hat{y}\big] = \int \hat{y}^T\hat{y}\; p(\hat{y}|z,\mathcal{D})d\hat{y} \tag{E.5}$$

$$= \int \Big(\int \hat{y}^T\hat{y}\; p(\hat{y}|\mathcal{W},z)d\hat{y}\Big)\; p(\mathcal{W}|\mathcal{D})d\mathcal{W} \tag{E.6}$$

$$= \int \big(\sigma^2 + \texttt{Net}(\mathcal{W},z)^2\big)\; p(\mathcal{W}|\mathcal{D})d\mathcal{W} \tag{E.7}$$

An unbiased estimate of the variance is given by Monte-Carlo integration methods [19,41], with M samples from the inferred posterior distribution of the network weights $\mathcal{W}$.

$$\mathbf{\Sigma}_{\hat{\mathbf{y}}} \approx \sigma^2 + \frac{1}{M}\sum_{m=1}^{M} \texttt{Net}(\mathcal{W}(m),z)^2 - {\mu_{\hat{\mathbf{y}}}}^T\mu_{\hat{\mathbf{y}}} \tag{E.8}$$

F Regularization Update Rules

To enforce a group regularization on network parameters, the prior formulation is revisited. The main difference is with the optimization step for ψ. Parameters in the same row share the prior uncertainty parameter $\psi^l_{a:}$ and in the same column the prior uncertainty $\psi^l_{b:}$.

For instance, the optimization step in equation (C.12) for $\psi^l_{:b}$, the prior width shared among the connection weights in the same column, becomes

$$\psi^l_{:b}(k+1) = \underset{\psi\geq 0}{\operatorname{argmin}} \sum_{b=1}^{n^l} \frac{W^l_{:b}(k+1)^T W^l_{:b}(k+1)}{\psi} + \alpha^l_{:b}(k)\cdot\psi \tag{F.1}$$

where $\alpha^l_{:b} = \sum_{a=1}^{n_{l-1}} \alpha^l_{ab}(k)$. By noting that

$$\sum_{b=1}^{n^l} \frac{{W^l_{:b}}^T W^l_{:b}}{\psi} + \alpha^l_{:b}\cdot\psi \geq 2\Big|\Big|\sqrt{\alpha^l_{:b}}\cdot W^l_{:b}\Big|\Big|_{l_2} \tag{F.2}$$

the analytical solution is given by $\psi^l_{:b}(k+1) = \frac{||W^l_{:b}(k+1)||_2}{\omega^l_{:b}(k)}$ where $\omega^l_{:b}(k) = \sqrt{\alpha^l_{:b}(k)} = \sqrt{\sum_{a=1}^{n_{l-1}} \alpha^l_{ab}(k)}$.

The row-wise regularization can be analogously derived. Note that the update rules for α^l_{ab} remains similar to equations (20) and (21). However when using both row-wise and column-wise group regularization, the posterior is updated according to a combined prior expressed with a prior width given by :

$$\psi^l_{ab}(k) = \frac{1}{(\frac{1}{\psi^l_{a:}(k)} + \frac{1}{\psi^l_{:b}(k)})} \tag{F.3}$$

G Hairdryer

In common industrial settings with heating, temperature control is a highly desired objective given the high transport lags and process delay. The "hairdryer" is a small scale laboratory apparatus that designates the PT326 process trainer [21]. A mass of air is heated with thermal resistors and flows in a tube. The temperature at the outlet is measured by a thermocouple in volts. The objective is to identify the dynamic relationship between the input voltage to the thermal resistors and the thermocouple voltage at the outlet. The dataset specific to this device is given by MATLAB in a tutorial on linear system identification. The sampling time is 0.08 seconds and the dataset contains 1000 data points. The dataset is detrended, bringing data to a zero mean. The first 300 data points are used for identification and the remaining 700 are used for validation.

A fully connected MLP model with one hidden layer and 50 nodes is randomly initialized. The activation function is a linear activation without the bias term. The input and output lags chosen for the regressors is 5. Models are inferred through $K_{\max} = 6$ identification cycles. The best validated model was obtained in the 5th cycle of identification with a sparsity of 88.1%. The model sparsity plot is shown in Fig. G.1. Furthermore, an RNN network is randomly initialized with one layer and 10 hidden LSTM units and no bias term. l_u and l_y are set to 5. The 6th and final identification cycle led to the sparsest and best validated model with a sparsity of 93.5 %. Fig. G.2 shows the final model sparsity plot.

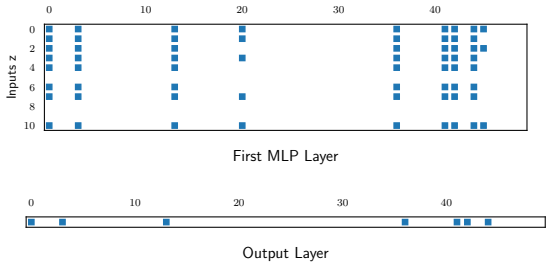


Fig. G.1. Model sparsity of the identified MLP on Hairdryer dataset. *Blue indicates non-pruned connections and white indicate pruned ones. The same follows with other sparsity plots.*

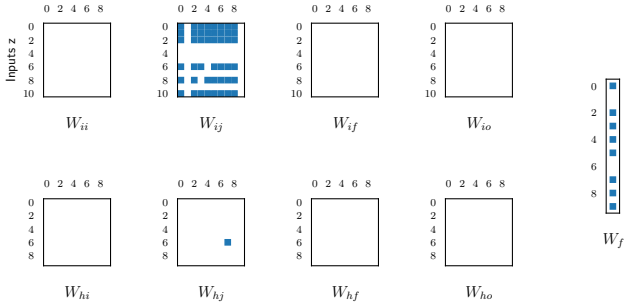


Fig. G.2. Model sparsity of the identified LSTM on Hairdryer dataset

Plots of the posterior predictive distribution's mean predictions and standard deviations obtained by sampling 10000 times from the posterior distribution of the connections' weights and by using equations (26) and (27) are shown in Fig. G.3 and G.4. Plots of the identified models' free run simulations can be found in Fig. M.1.

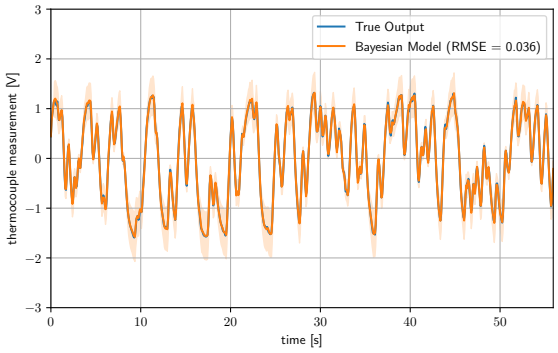


Fig. G.3. Posterior mean predictions of the identified MLP on Hairdryer dataset ($\pm 2\sigma$)

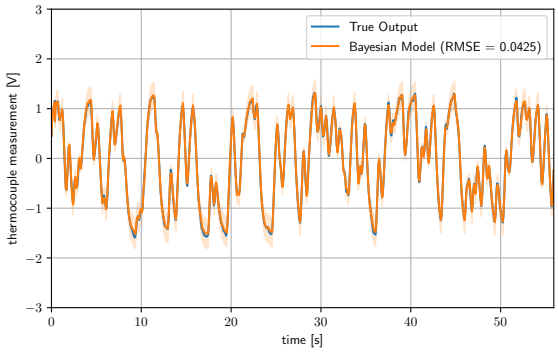


Fig. G.4. Posterior mean predictions of the identified LSTM on Hairdryer dataset ($\pm 2\sigma$)

H Heat Exchanger

A heat exchanger is a thermodynamic device that ensures a transfer of heat in between two fluids seperated by a wall. In this experiment, the dynamic relationship between the change in coolant temperature and the change in the product temperature is identified [21]. The first 3000 data points are used for identification and the remaining 2000 for validation. This dataset is particularly unique among the others. The process exhibits a delay of around 1/4 of a minute [21].

One hidden-layer MLP with 50 nodes is initialised with a linear activation function and no bias term. The lag chosen is $l_u = l_y = 150$ samples corresponding to the delay of 0.25 seconds that can be observed in the first instances of the given dataset. The experiment ran for 6 identification cycles, in which the 4th obtained model was selected as the best validated model. The model is 99.3% sparse .

One layer RNN network with 10 LSTM units is trained with the same lag used previously ($l_u = l_y = 150$). The best validated model was the second out of 6 identification cycles. The accepted model's sparsity is 96.4 % for which the sparsity plot is given in Fig. H.2.

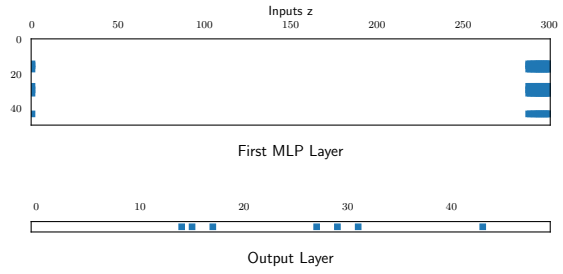


Fig. H.1. Model sparsity of the identified MLP on Heat Exchanger dataset.

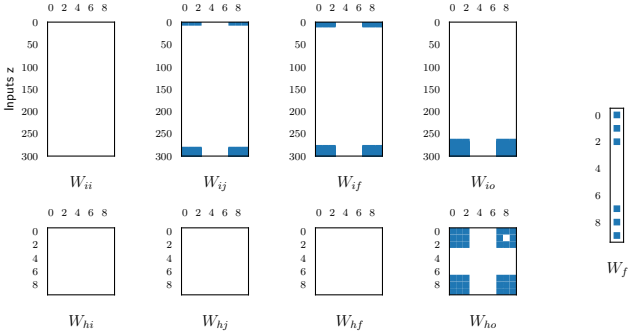


Fig. H.2. Model sparsity of the identified LSTM on Heat Exchanger dataset.

The predictive mean and standard deviation of the posterior predictive distribution are shown in Fig. H.3-H.4 against the real validation signal. These were obtained using 10000 samples of the posterior distributions. Please refer to Fig. M.2, for a plot of these free run simulations.

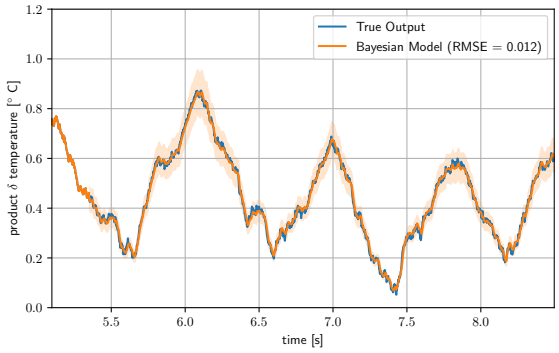


Fig. H.3. Posterior mean predictions of the identified MLP on Heat Exchanger dataset ($\pm 2\sigma$)

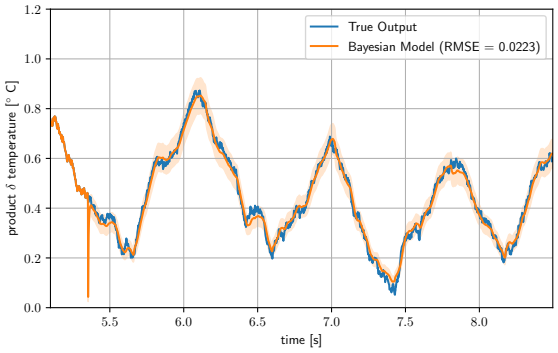


Fig. H.4. Posterior mean predictions of the identified LSTM on Heat Exchanger dataset ($\pm 2\sigma$)

I Glass Tube Manufacturing Process

In the process of manufacturing glass tubes, melted glass shapes around a rotating cylinder, while homogenizing. It is, then drawn on rollers to a certain length. The thickness of the obtained glass tube is measured by a laser beam outside the chamber [42]. The objective is to identify the linear dynamic relationship between the input drawing speed and the output thickness. The datasets are provided by the MATLAB example. These are detrended and decimated by four, to get rid of high frequency components of the signal [22]. This results in a sampling time of 4 seconds. The data used for identification consist of the first 500 datapoints and the remaining is used in validation.

An MLP is randomly initialized with one hidden layer and 50 neurons. The input regressors are chosen such as $l_u = l_y = 5$. The activation function used is linear without a bias term. The final obtained model is 97.8 % sparse with a sparsity plot shown in Fig. I.1. This model was the third generated model out of 6 identification cycles.

With the same choice of regressors, an RNN network was initialized with one layer of 10 LSTM units. The bias term was not used in this case. In the 6 identification cycles, the 6th generated model was the sparsest and have the best validation performance. The sparsity plot of this network is given by Fig. I.2. The model is 99% sparse, and the only non-pruned parameters in the model correspond to the input to cell state operator W_{ij}.

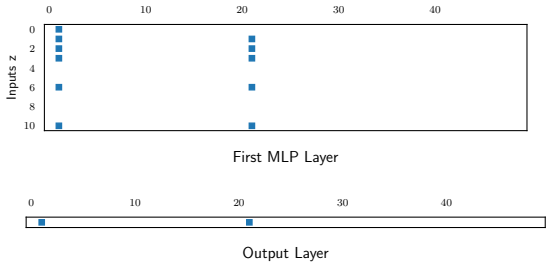


Fig. I.1. Model sparsity of the identified MLP on Glass Tube Manufacturing dataset.

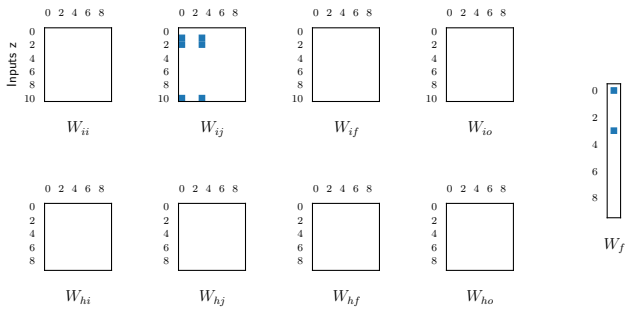


Fig. I.2. Model sparsity of the identified LSTM on Glass Tube Manufacturing dataset.

The one-step ahead prediction estimates and uncertainties are obtained by Monte Carlo sampling 10000 times from the posterior and are shown in Fig. I.3-I.4 as a representation of the posterior predictive distribution. The free run simulations of the generated models in this paper are presented in Fig. M.3.

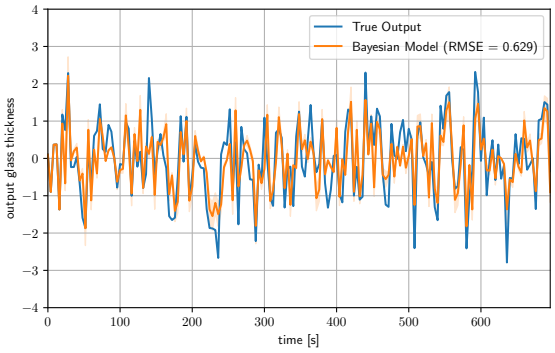


Fig. I.3. Posterior mean predictions of the identified MLP on Glass Tube Manufacturing dataset ($\pm 2\sigma$)

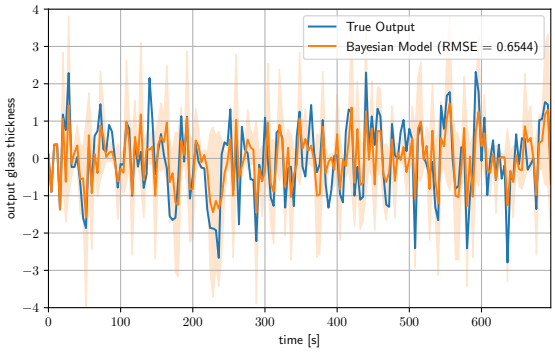


Fig. I.4. Posterior mean predictions of the identified LSTM on Glass Tube Manufacturing dataset ($\pm 2\sigma$)

J Cascaded Tanks

A pump drives water up from a reservoir to the upper tank of two vertically cascaded tanks. The upper and lower tanks are seperated by a small opening allowing water to fill the lower tank. The lower tank and the reservoir are also seperated by a small opening, from which water goes back to the reservoir. In addition to that, water can overflow from the upper tank to the lower tank and reservoir. Water can also overflow the second tank and drop into the reservoir. The small openings and overflows are sources of non-linearity [24]. The objective of the benchmark is the identification of the dynamic relationship between the input voltage to the pump and the output measured water level in the lower tank by a capacitive sensor [24]. Two multisine input datasets and their corresponding outputs with a sampling rate of 4 seconds are provided. The datasets contain each 1024 samples and are with different initial conditions. One of the datasets is used for estimation and the other for validation. The signals provided exhibit a static bias that is dealt with in the pre-processing stage of the identification procedure by detrending.

A 3 hidden layers deep MLP network with 10 neurons per layer is randomly initialized. The activation function used is the relu activation. The input regressors are such as $l_u = l_y = 20$. The identification experiment is ran for 10 cycles. The 9th generated model pig erforms the best in validation with a sparsity of 84.5%. The model's sparsity plot is shown in Fig. J.1.

Moreover, a one layer RNN with 10 LSTM units is also used as a model structure for the identification experiment. The 4th identified model with 60.3 % sparsity was the best validated model out of 10 identification cycles. The sparsity plot of the corresponding model is shown in Fig. J.2.

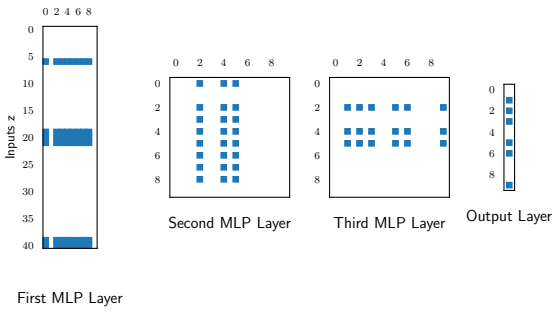


Fig. J.1. Cascaded Tanks MLP Model sparsity plot

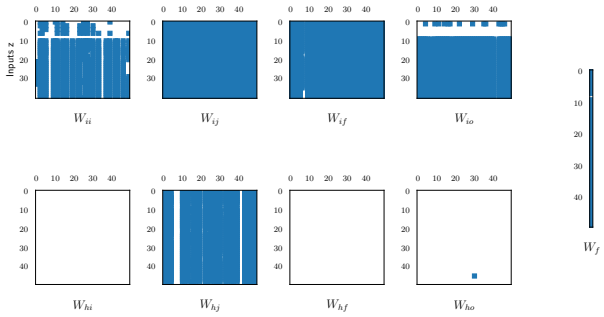


Fig. J.2. Cascaded Tanks RNN Model sparsity plot

In addition, the posterior predictive mean and standard deviation are given in Fig. J.3 and J.4. These were obtained by the averaging equations 26 and 27 and sampling 50000 times from the inferred posterior distribution of the weights. A plot of the models' free run simulations is by Fig. M.4.

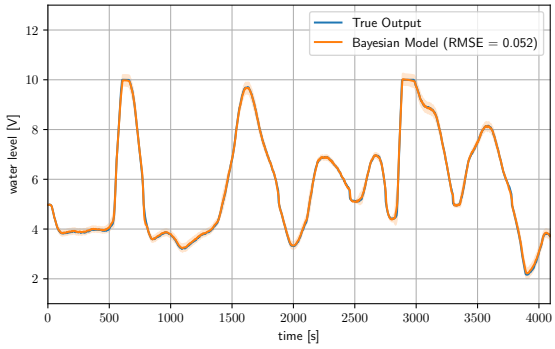


Fig. J.3. Cascaded Tanks MLP models' output posterior mean predictions ($\pm 2\sigma$)

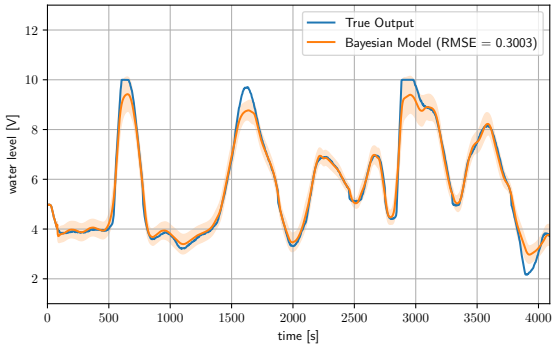


Fig. J.4. Cascaded Tanks LSTM models' output posterior mean predictions ($\pm 2\sigma$)

K Coupled Electric Drives

The coupled electric drives consists of 2 electric motors and a pulley, connected by a flexible belt forming a triangle. The pulley is attached by a spring to a fixed frame. This results in belt tension, slippage and pulley speed, that is harder to model. In addition to that, the output pulley rotational speed is measured in ticks per seconds, insensitive to rotational directions. The dynamic relationship to be identified is between the input motors voltage and the measured rotational speed of the pulley. For this identification task, 2 uniformly distributed signals of 500 samples is provided spanning 10 seconds. With each of these datasets, the first 300 samples are used for estimation and the remaining for validation.

Two hidden layers MLP with 50 neurons each and relu activation functions is randomly initialized and trained with the estimation data for 10 identification cycles. The model's regressors are chosen such that $l_u = l_y = 10$. The model obtained in the 6th identification iteration is the chosen best model. This model is 78.4% sparse for which the sparsity plot is shown in Fig. K.1.

The same regressors are used for the identification using RNN model structure. An RNN with one layer and 10 LSTM units is trained for 10 identification cycles. The 8th identification yields the best simulation validation results. The resulting model is 72.8% sparse with the sparsity plot in Fig. K.2.

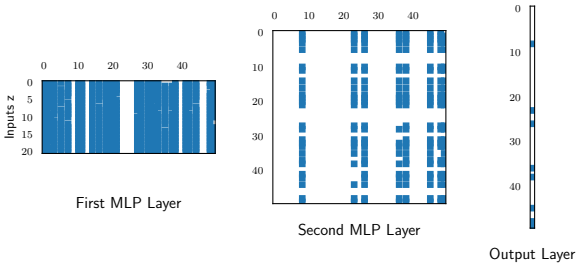


Fig. K.1. Coupled Electric Drives MLP Model sparsity plot

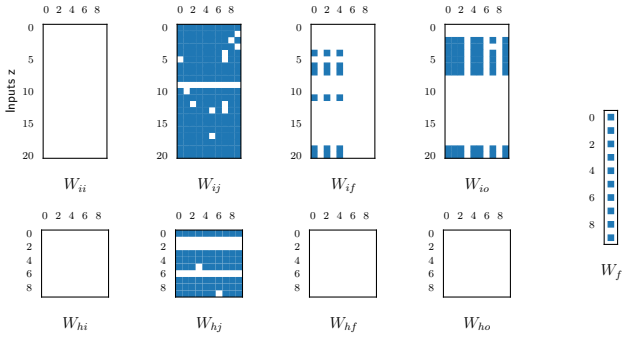


Fig. K.2. Coupled Electric Drives RNN Model sparsity plot

By using equations 26 and 27, the mean and standard deviation of the posterior predictive distributions is plotted in Fig. K.3, K.5, K.4 and K.6 for both validation datasets. These are obtained with equations (26)-(27) and 50000 samples of the posterior distribution. Figures showing the resulting free run simulations are Fig. M.5-M.6.

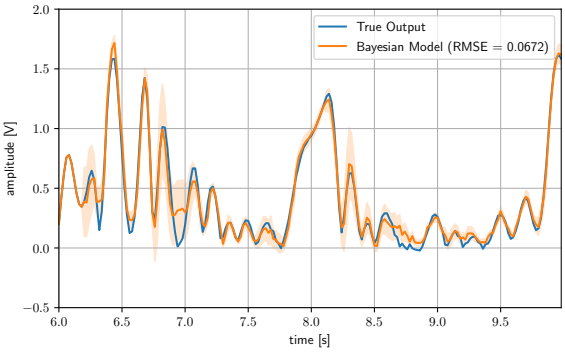


Fig. K.3. Coupled Electric Drives MLP models' output posterior mean predictions ($\pm 2\sigma$) of first validation dataset

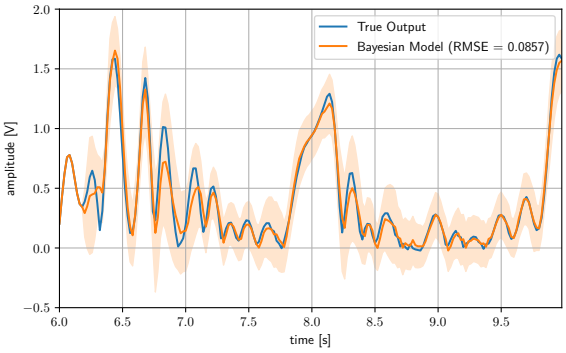


Fig. K.4. Coupled Electric Drives RNN models' output posterior mean predictions ($\pm 2\sigma$) of first validation dataset

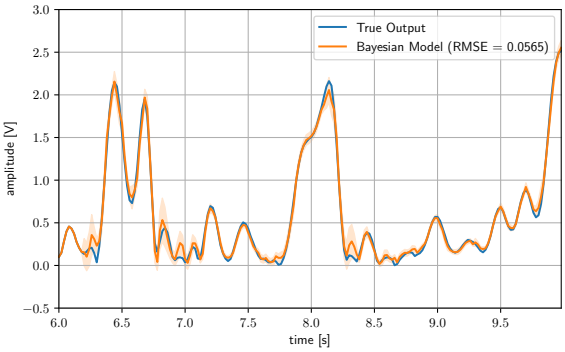


Fig. K.5. Coupled Electric Drives MLP models' output posterior mean predictions ($\pm 2\sigma$) of second validation dataset

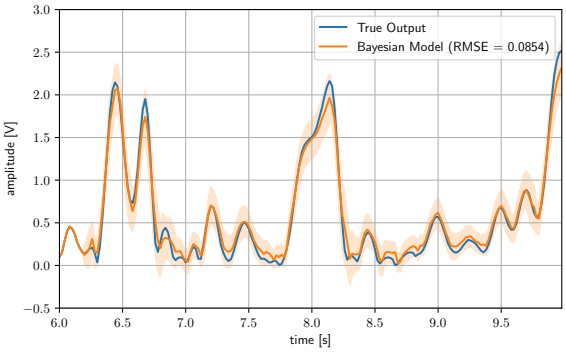


Fig. K.6. Coupled Electric Drives RNN models' output posterior mean predictions ($\pm 2\sigma$) of second validation dataset

L Bouc-Wen Hysteresis Model

Hysteresis is a complex non-linear dynamic phenomenon observed in vibration science, materials, magnetism and many other applications in both physical and social sciences. It is characterized by a dynamic system's dependency for previous states and is challenging to model mathematically. The Bouc-Wen model for hysteresis is one of the most versatile parametric models used in a wide range of hysteric applications [43]. The model relates the hysteric restoring force (in N) to displacement (in mm). A Matlab simulink code is provided alongside two validation signals: a random multisine and a sinesweep [44]. The signal used for estimation is five realizations of 8192 multisine samples with a sampling frequency of 750 Hz and an additive band limited Gaussian noise in 0-375 Hz. Both inputs and outputs have very different decimal means and scales, hence, the datasets are normalized before estimation.

A first identification experiment is done with an MLP network of 2 hidden layers, 50 neurons per layer and relu activation functions. The identification procedure includes 10 iterations, in which the best model generated was the 9th. This model is 38.5% sparse. A visual representation of this sparse model is Fig. L.1.

The second identification experiment is done with an RNN composed of a layer of 10 LSTM units. The final accepted identified model is the 9th model with a sparsity of 82.8 %. Due to the limitation of resources, only the first 10% of the estimation data is used for identification using LSTM units. Fig. L.2 is a visual representation of the obtained model sparsity.

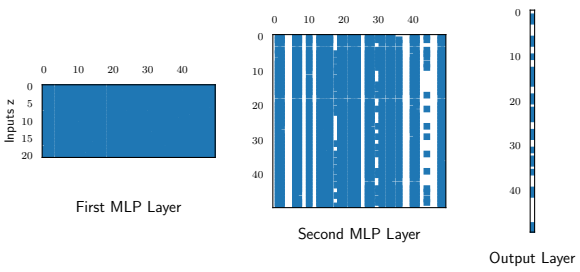


Fig. L.1. Bouc-Wen MLP Model sparsity plot

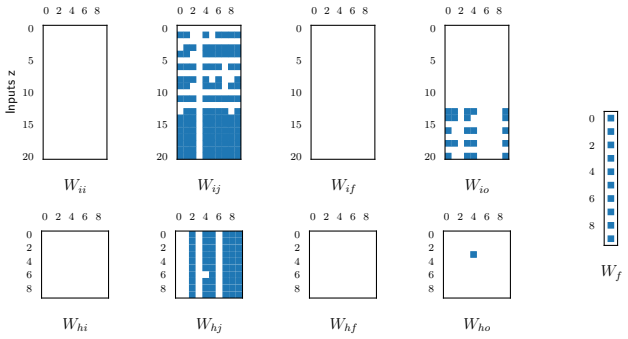


Fig. L.2. Bouc-Wen RNN Model sparsity plot

Similarly to other identified processes, the mean and uncertainty estimates in model's prediction is shown for both validation datasets on a small window of a 1000 samples. For the first validation data, the window spans samples 1000 to 2000 and the plot is shown in Fig. L.3 and L.4. For the second validation dataset, the chosen window spans from

sample 4000 and 5000, plotted in Fig. L.5 and L.6. To obtain these, the posterior was sampled for 50000 times. Plots of the free run simulations are given by Fig. M.7-M.8.

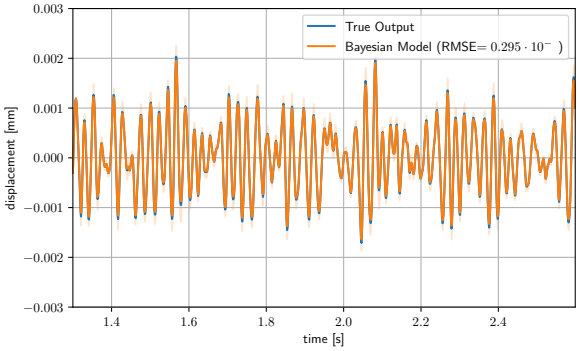


Fig. L.3. Bouc-Wen MLP models' output posterior mean predictions ($\pm 2\sigma$) of first validation dataset

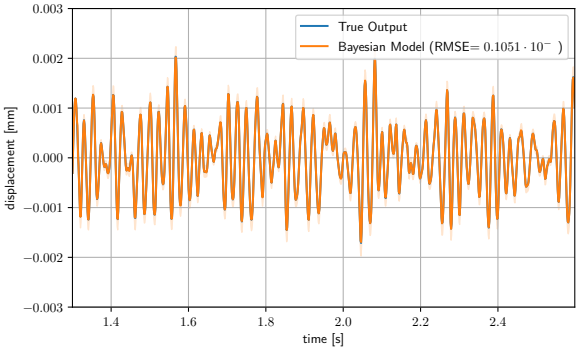


Fig. L.4. Bouc-Wen RNN models' output posterior mean predictions ($\pm 2\sigma$) of first validation dataset

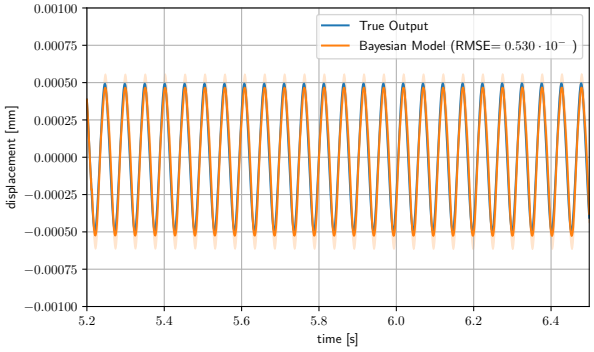


Fig. L.5. Bouc-Wen MLP models' output posterior mean predictions ($\pm 2\sigma$) of second validation dataset

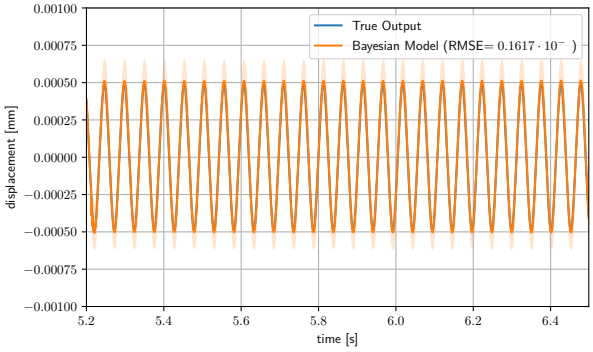


Fig. L.6. Bouc-Wen RNN models' output posterior mean predictions ($\pm 2\sigma$) of second validation dataset

M Free Run Simulation Plots

This section of the appendix supports the reader with plots of the simulated experiments using the models identified.

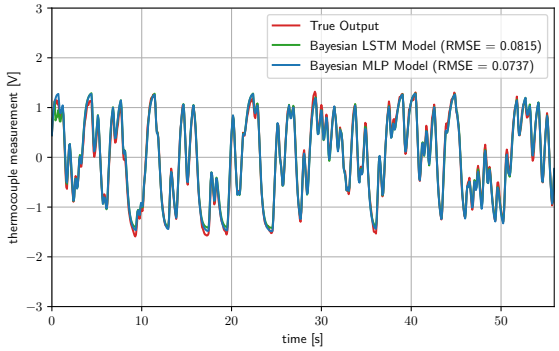


Fig. M.1. Hairdryer Free Run Simulation comparison

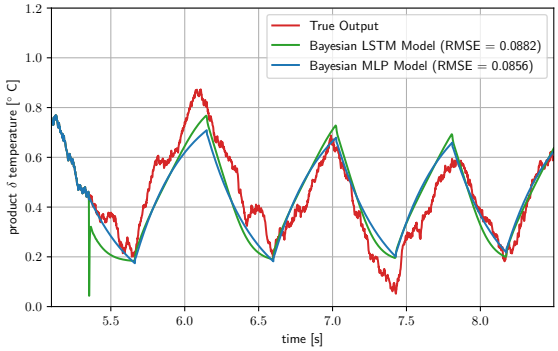


Fig. M.2. Heat Exchanger Free Run Simulation comparison

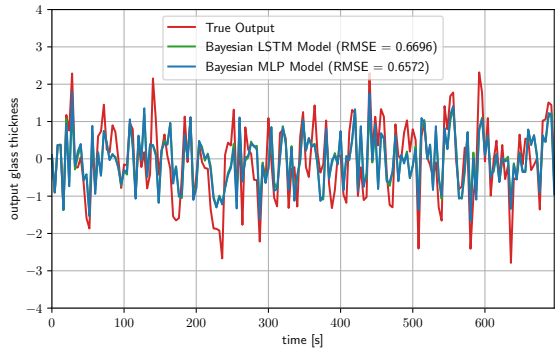


Fig. M.3. Glass Tube Manufacturing Free Run Simulation comparison

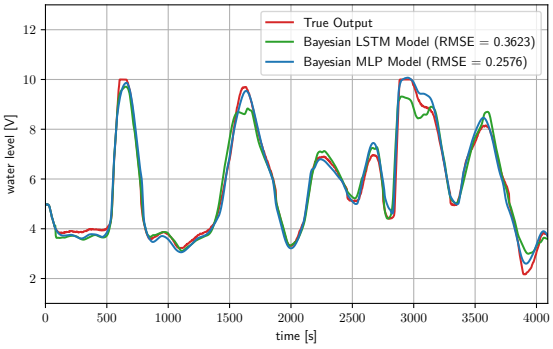


Fig. M.4. Cascaded Tanks Free Run Simulation comparison

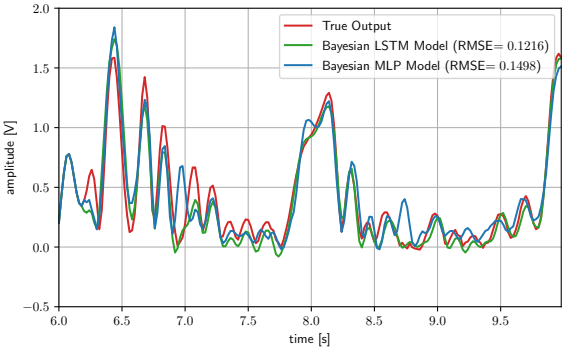


Fig. M.5. Coupled Electric Drives Free Run Simulation comparison for the first validation dataset

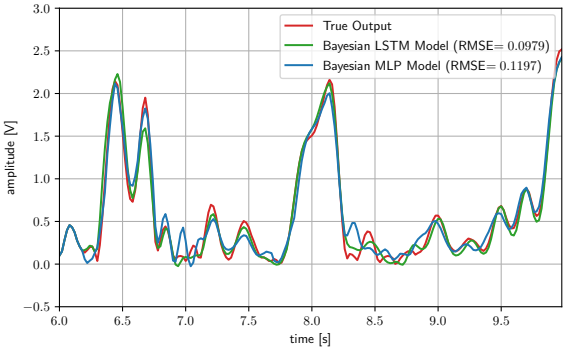


Fig. M.6. Coupled Electric Drives Free Run Simulation comparison for the second validation dataset

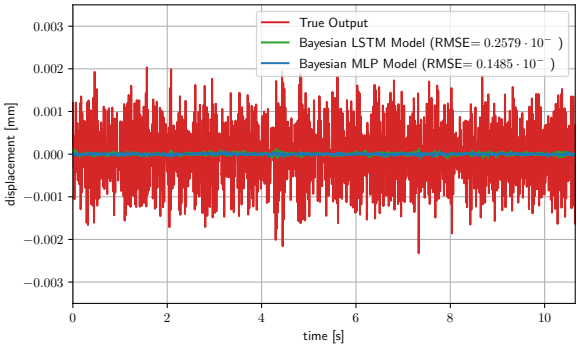


Fig. M.7. Bouc-Wen Free Run Simulation error comparison for the first validation dataset

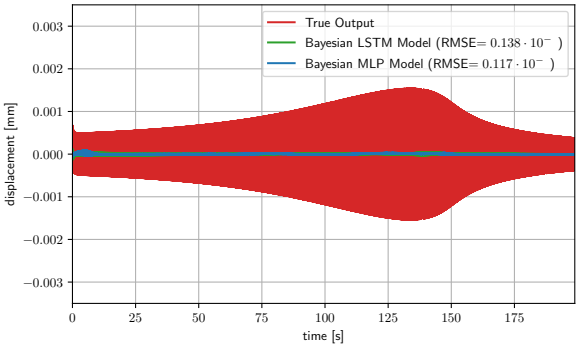


Fig. M.8. Bouc-Wen Free Run Simulation error comparison for the second validation dataset

Chapter 4

Conclusion

4-1 Summary

Bayesian Neural Networks are a good choice of methods to identify dynamic systems. In this work, linear and non-linear processes were identified using Multi-Layer Perceptrons and Long Short Term Memory in a Bayesian Approach. Sparsity inducing group priors are introduced and the Laplace approximation is used to approximate the evidence. The type II maximum likelihood problem (evidence maximization) is recast into an iterative l_1 procedure using the Concave-Convex-Procedure. In addition, with the inferred posterior distribution of the connection weights, the predictions made by the network can be obtained in a distribution form rather than only point estimates by using Monte Carlo Integration methods. Six processes were identified to demonstrates sparsity, predictive uncertainty quantification and competitive simulation validation results with previous works found in literature. The work done for this dissertation was compiled in a journal paper that also constitutes the body of this document.

4-2 Limitations

The presented Bayesian Deep Learning model and algorithm exhibit current limitations. These include:

- The Laplace approximation adopted requires computing the Hessian of the log likelihood with respect to the connection weights as well as an expression of its inverse for the posterior. This may be infeasible in the case of large networks and incurs a high computational cost. In addition, the convergence properties remain local even when these are improved with the heuristics introduced such as pruning and adaptive regularization. Nonetheless, the method's computational costs, tuning costs and convergence properties can well improve with advancement in training neural networks and Hessian matrix approximation methods.

- The uncertainty in model parameters and predictions are uncertainties related to a fit on estimation data. The Bayesian approach attempts to balance a model fit and complexity based on the estimation data and hence improves generalization property. However, similar to many identification methods, the quality of estimation data remains very influential to the model fit and hence to the quantified uncertainties in predictions. In the next subsection, we shall see how this limitation can also be an opportunity.
- The Bayesian approach is often seen as an attempt to demystify deep learning. That was also seen in this thesis. The Bayesian method combined with the fitting capabilities of artificial neural networks provided insights such as quantified uncertainties and, in some cases, an automatic selection of regressors. However, models remain opaque (especially for non-linear model estimation) and parameters inferred do not represent any physical quantity. Furthermore, the prior form is subjective, and does not include any prior knowledge of modeling and dynamic system analysis.

4-3 Opportunities

Some research directions to explore in the future include:

- **Extension to MIMO systems:** The Bayesian method used can be extended to multiple-input multiple-output processes by simply extending the network inputs and outputs with the system's inputs and measurements. It might also be interesting to start with a small model (one intermediate layer), where direct relationships and coupling between inputs and outputs can be better inferred by effect of the structural sparsity imposed.
- **Uncertainty and data acquisition:** The posterior predictive uncertainty is a reflection of how inferred model parameters reflect in uncertainty to predictions. In other words, if in the validation phases an input in a certain window of time generated a high uncertainty in the posterior predictive distribution, this input is highly informative. According to these validation results, users can redesign input properties related to frequency, amplitude or type of signal and re-acquire estimation data for another iteration of system identification. An analogous view can be found in reinforcement learning, where uncertainty can help agents with the balance between exploration and exploitation.
- **More prior information:** Given the physics of a problem, the network structure can be designed to include more prior information. This can be done by designing a prior network that would be integrated with a fully connected network. For instance, in robotics, forward kinematics often involve trigonometric functions to map rotational actuation to end effector position/speed. In this case, prior network would attempt to recover this trigonometric relationship and the fully connected network would compensate for imperfections and interference. An example of such units would be linear/non-linear combinations of trigonometric functions found in the forward model. It is envisioned that the Bayesian approach would render a highly sparse fully connected network and help quantify uncertainty in the inferred parameters of the prior and fully connected units.

Bibliography

[1] H. V. H. Ayala, L. F. da Cruz, R. Z. Freire, and L. dos Santos Coelho. Cascaded free search differential evolution applied to nonlinear system identification based on correlation functions and neural networks. In *2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, pages 1–7, Dec 2014.

[2] D. Barber and Christopher Bishop. Ensemble learning in bayesian neural networks. In *Generalization in Neural Networks and Machine Learning*, pages 215–237. Springer Verlag, January 1998.

[3] Julian Belz, Tobias Münker, Tim O. Heinz, Geritt Kampmann, and Oliver Nelles. Automatic modeling with local model networks for benchmark processes. *IFAC-PapersOnLine*, 50(1):470 – 475, 2017. 20th IFAC World Congress.

[4] Christopher M. Bishop and Michael E. Tipping. Variational relevance vector machines. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, UAI'00, page 46–53, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[5] Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical gauss-newton optimisation for deep learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 557–565. JMLR.org, 2017.

[6] Mathieu Brunot, Alexandre Janot, and Francisco Javier Carrillo. Continuous-time nonlinear systems identification with output error method based on derivative-free optimisation. In *IFAC World Congress 2017*, Toulouse, FR, July 2017.

[7] Peter Buhlmann and Sara van de Geer. *Theory for l1/l2-penalty procedures*, pages 249–291. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[8] Eduardo F. Camacho and Carlos Bordons. *Model Based Predictive Controllers*, pages 13–31. Springer London, London, 1999.

[9] Petr Chalupa, Jakub Novak, and Michal Jarmar. Model of coupled drives apparatus – static and dynamic characteristics. *MATEC Web of Conferences*, 76:02011, 01 2016.

[10] Alireza Fakhrizadeh Esfahani, Philippe Dreesen, Koen Tiels, Jean-Philippe Noël, and Johan Schoukens. Polynomial state-space model decoupling for the identification of hysteretic systems. *IFAC-PapersOnLine*, 50(1):458 – 463, 2017. 20th IFAC World Congress.

[11] Anita C. Faul and Michael E. Tipping. Analysis of sparse bayesian learning. In *NIPS*, 2001.

[12] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective, 2020.

[13] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[14] X. He and H. Asada. A new method for identifying orders of input-output models for nonlinear dynamic systems. In *1993 American Control Conference*, pages 2520–2523, 1993.

[15] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366, 1989.

[16] Roland Hostettler, Filip Tronarp, and Simo Särkkä. Modeling the drift function in stochastic differential equations using reduced rank gaussian processes. *IFAC-PapersOnLine*, 51(15):778–783, 2018. 18th IFAC Symposium on System Identification SYSID 2018.

[17] Yong Huang, Changsong Shao, Biao Wu, James L. Beck, and Hui Li. State-of-the-art review on bayesian inference in structural system identification and damage assessment. *Advances in Structural Engineering*, 22(6):1329–1351, 2019.

[18] The MathWorks Incorporation. Estimating simple models from real laboratory process data. https://nl.mathworks.com/help/ident/ug/estimating-simple-models-from-real-laboratory-process-data.html. Accessed: 2020-11-25.

[19] The MathWorks Incorporation. Estimating transfer function models for a heat exchanger. https://nl.mathworks.com/help/ident/ug/estimating-transfer-function-models-for-a-heat-exchanger.html. Accessed: 2020-11-25.

[20] The MathWorks Incorporation. Glass tube manufacturing process. https://nl.mathworks.com/help/ident/ug/glass-tube-manufacturing-process.html. Accessed: 2020-11-25.

[21] The MathWorks Incorporation. System identification toolbox — examples. https://nl.mathworks.com/help/ident/examples.html. Accessed: 2020-11-25.

[22] Mohammed Ismail, Fayçal Ikhouane, and José Rodellar. The hysteresis bouc-wen model, a survey. *Archives of Computational Methods in Engineering*, 16:161–188, 06 2009.

[23] Rios J., Alanis A., Arana-Daniel N., and Lopez-Franco C. *Neural Networks Modeling and Control*. Academic Press, 1st ed. 2020. edition.

[24] Ridvan Karagoz and Kim Batselier. Nonlinear system identification with regularized tensor network b-splines. *Automatica*, 122, 2020.

[25] Diederik Kingma and Max Welling. Auto-encoding variational bayes. 12 2014.

[26] Hari Koduvely. *Learning Bayesian Models with R*. 10 2015.

[27] Gert R. Lanckriet and Bharath K. Sriperumbudur. On the convergence of the concave-convex procedure. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1759–1767. Curran Associates, Inc., 2009.

[28] Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 598–605. Morgan-Kaufmann, 1990.

[29] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

[30] Lennart Ljung. *System Identification*, chapter 5.4. American Cancer Society, 1999.

[31] Lennart Ljung. *System Identification (2nd Ed.): Theory for the User*. Prentice Hall PTR, USA, 1999.

[32] Lennart Ljung, Carl Andersson, Koen Tiels, and Thomas B. Schön. Deep learning and system identification. In *Deep learning and system identification*, February 2020.

[33] David J. C. MacKay. *Bayesian Interpolation*, pages 39–66. Springer Netherlands, Dordrecht, 1992.

[34] David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Comput.*, 4(3):448–472, May 1992.

[35] Per Mattsson, Dave Zachariah, and Petre Stoica. Identification of cascade water tanks using a pwarx model. *Mechanical Systems and Signal Processing*, 106:40 – 48, 2018.

[36] Rowan McAllister, Yarin Gal, Alex Kendall, Mark van der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4745–4753, 2017.

[37] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, March 1990.

[38] Radford M. Neal. *Introduction*, pages 1–28. Springer New York, New York, NY, 1996.

[39] Stefan-Cristian Nechita, Roland Toth, Dhruv Khandelwal, and Maarten Schoukens. Toolbox for discovering dynamic system relations via tag guided genetic programming, 2020.

[40] H. Nejib, O. Taouali, and N. Bouguila. Identification of nonlinear systems with kernel methods. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 000577–000581, Oct 2016.

[41] Oliver Nelles. *Dynamic Neural and Fuzzy Models*, page 588. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[42] Oliver Nelles. *Nonlinear Dynamic System Identification*, pages 547–577. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[43] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, 2019.

[44] Jason Palmer, Kenneth Kreutz-Delgado, Bhaskar D. Rao, and David P. Wipf. Variational em algorithms for non-gaussian latent variable models. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1059–1066. MIT Press, 2006.

[45] Jason Palmer, Bhaskar D. Rao, and David P. Wipf. Perspectives on sparse bayesian learning. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 249–256. MIT Press, 2004.

[46] W. Pan, Y. Yuan, J. Gonçalves, and G. Stan. A sparse bayesian approach to the identification of nonlinear state-space systems. *IEEE Transactions on Automatic Control*, 61(1):182–187, Jan 2016.

[47] Wei Pan. *Bayesian Learning for Nonlinear System Identification*. PhD dissertation, Imperial College London, 2017.

[48] Duc T. Pham. *Neural Networks for Identification, Prediction and Control*. Springer London, London, 1st ed. 1995. edition.

[49] Carlos Ramirez. Why l1 is a good approximation to l0: A geometric explanation. *Journal of Uncertain Systems*, 7:203–207, 08 2013.

[50] Rishi Relan, Koen Tiels, Anna Marconato, and Johan Schoukens. An unstructured flexible nonlinear model for the cascaded water-tanks benchmark. *IFAC-PapersOnLine*, 50(1):452–457, 2017. 20th IFAC World Congress.

[51] Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016.

[52] F. Sabahi and M. R. Akbarzadeh-T. Extended fuzzy logic: Sets and systems. *IEEE Transactions on Fuzzy Systems*, 24(3):530–543, June 2016.

[53] Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, Jun 2017.

[54] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini. Novel cascade spline architectures for the identification of nonlinear systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(7):1825–1835, July 2015.

[55] Anton Maximilian Schäfer and Hans Georg Zimmermann. Recurrent neural networks are universal approximators. In *Proceedings of the 16th International Conference on Artificial Neural Networks - Volume Part I*, ICANN'06, page 632–640, Berlin, Heidelberg, 2006. Springer-Verlag.

[56] Jonathan Schaffer. What not to multiply without necessity. *Australasian Journal of Philosophy*, 93:1–21, 05 2014.

[57] M. Schoukens, Per Mattsson, Torbjörn Wigren, and J.M.M.G. Noël. Cascaded tanks benchmark combining soft and hard nonlinearities. In *Workshop on Nonlinear System Identification Benchmarks : April 25-27, 2016, Brussels, Belgium*, pages 20–23, April 2016. 2016 Workshop on Nonlinear System Identification Benchmarks ; Conference date: 25-04-2016 Through 27-04-2016.

[58] M. Schoukens and J.P. Noël. Three benchmarks addressing open challenges in nonlinear system identification. *IFAC-PapersOnLine*, 50(1):446 – 451, 2017. 20th IFAC World Congress.

[59] Maarten Schoukens and Fritjof Griesing Scheiwe. Modeling nonlinear systems using a volterra feedback model. In *Workshop on Nonlinear System Identification Benchmarks*, 2016.

[60] H.T. Siegelmann and E.D. Sontag. On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1):132–150, 1995.

[61] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[62] Shiliang Sun. A review of deterministic approximate inference techniques for bayesian machine learning. *Neural Computing and Applications*, 23:2039–2050, 2013.

[63] Andreas Svensson and Thomas B. Schön. A flexible state–space model for learning nonlinear dynamical systems. *Automatica*, 80:189–199, 2017.

[64] Hong Hui Tan and King Hann Lim. Review of second-order optimization techniques in artificial neural networks backpropagation. 2019.

[65] Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, September 2001.

[66] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos. The variational approximation for bayesian inference. *IEEE Signal Processing Magazine*, 25(6):131–146, 2008.

[67] Michel Verhaegen and Vincent Verdult. *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, USA, 1st edition, 2007.

[68] V. Wertz, G. Bastin, and M. Haest. Identification of a glass tube drawing bench. *IFAC Proceedings Volumes*, 20(5, Part 10):333–338, 1987. 10th Triennial IFAC Congress on Automatic Control - 1987 Volume X, Munich, Germany, 27-31 July.

[69] T. Wigren and M. Schoukens. *Coupled electric drives data set and reference models*. Number 024 in Technical Report Uppsala Universitet. Uppsala University Sweden, November 2017.

[70] Andrew Gordon Wilson. The case for bayesian deep learning, 2020.

[71] D. P. Wipf and B. D. Rao. Sparse bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, 52(8):2153–2164, 2004.

[72] Tiumentsev Y. and Egorchev M. *Neural Network Modeling and Identification of Dynamical Systems*. Academic Press, 1st ed. 2019. edition.

[73] A. L. Yuille and Anand Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

[74] Hongpeng Zhou, Minghao Yang, Jun Wang, and Wei Pan. Bayesnas: A bayesian approach for neural architecture search. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, pages 13116–13140. International Machine Learning Society (IMLS), 2019. 36th International Conference on Machine Learning, ICML 2019 ; Conference date: 09-06-2019 Through 15-06-2019.

Appendix

This chapter aims to provide an overview of the identified models in literature for three linear systems as well as three non-linear systems benchmarks. This section is part of the literature compiled for this work, but to avoid repetitions in the flow of the report, these are appended here if authors are interested in gaining more insight.

Each of these benchmarks is described starting with the mathematical model showing some physical intuition into the system and the provided input signals for estimation and validation. These would allow to deduce some challenges one may be faced with the identification experiment. Finally, previous literature works on these benchmarks are outlined in tables.

4-4 Appendix A: Benchmarks Description

Note that the terms one-step ahead prediction and free run simulation will be frequently mentioned in this thesis. A one-step prediction is an experiment where the model inputs can depend on the previous real outputs, while model input of a simulation cannot. Simulation inputs can, however, utilize previously estimated outputs by the same model.

4-4-1 Heat Exchanger

The heat exchanger is a family of systems found in some electronic devices, industrial engines or even households. These allow the heat transfer between at least two fluids physically separated by a barrier to avoid mixing. The dataset is a MATLAB dataset used as a demo in the system identification toolbox with the command `load iddemo_heatexchanger_data` [21].

Modeling and Physical Insight

The tutorial on the identification aims to identify the relationship between the change in coolant temperature and the change in product temperature in a heat exchanger around nominal values. Heat transfer exhibits a transient behavior when thermal properties are changed. In other words, a step change in coolant temperature does not trigger a constant flow of energy in the system and the outlet product temperature changes with respect to time until steady state. In an attempt to give more insight into a heat exchanger's dynamics, a model can be obtained by using the energy balance equation. Assuming constant fluid properties, the rate of thermal energy accumulation in the product fluid is given by:

$$\dot{E}_{acc} = \dot{E}_{in} - \dot{E}_{out} - \dot{E}_{loss} \tag{4-1}$$

$$m_p c_p \frac{d(\delta T_p)}{dt} = \dot{m}_p c_p \ \delta T_p + \dot{m}_c c_c \ \delta T_c - \dot{E}_{loss}(T_a) \tag{4-2}$$

With $\delta T = T_{in} - T_{out}$, T being the temperature of the coolant (subscript is $_c$), the product (subscript is $_p$) or the ambient (subscript is $_a$). c_c, c_p represent the heat capacities of respectively the coolant and the product fluid. m is the fluid mass and $\dot{m}$ the mass flow rate.

Given equation 4-2, the relationship between product and coolant temperature in a heat exchanger can be seen as a first order dynamic model with a disturbance term from ambient temperature. This was also mentioned in the tutorial for the identification of the heat exchanger dataset [19].

Benchmark Data Description

One set of inputs-outputs is used in the tutorial. In this set, the input signal consists of a pulse wave of different widths. It is important to note that an input-output delay of 0.25 seconds can be seen in the provided dataset. The signals contains 5000 samples and a sampling time of 0.0017 seconds. The first 3000 samples are used for estimation and the remaining 2000 for validation. The inputs are given by figures 4-1 and 4-2.

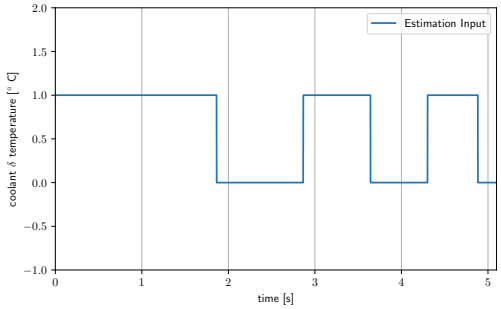


Figure 4-1: Heat Exchanger estimation input data.

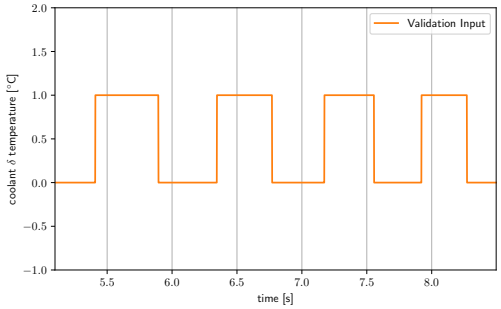


Figure 4-2: Heat Exchanger validation input data.

Identification Challenges

The main challenge in the identification of the heat exchanger model is the detected long delay (around 147 samples) and the disturbance of the ambient temperature for which no data is provided.

Previous Identification Works

Table 4-1: Comparison of other models from literature on Heat Exchanger Dataset

Previous Work	Method	Tests	RMSE [° C]
[19] The Mathworks Incorporation	Initialized Transfer Function Estimation `tfest(data, sysInit)`	One-Step Ahead Prediction Free Run Simulation	0.140
[19] The Mathworks Incorporation	Process Model Estimation `procest(data, "P1D")`	Free Run Simulation	0.88
[19] The Mathworks Incorporation	Process Model Estimation with ARMA Disturbance Model `procest(data, sysInit, opt)`	Free Run Simulation	0.89

4-4-2 Glass Tube Manufacturing

Manufacturing glass tubes includes three main processes, homogenization, forming and cooling. Melted glass is fed around a rotating cylinder to shape and homogenize glass in a cylindrical shape. A channel in through the center of the cylinder allows air to be blown into the forming zone of the process, where the glass takes a bulb shape. The end of the bulb is finally drawn given an input drawing speed and cooled into a final tube [68]. The summarized process is shown in figure 4-3. In this thesis, the goal is to identify the dynamic

model relating the drawing speed and tube thickness. The data used in the identification is provided in Matlab using the command `load thispe25.mat` [20].

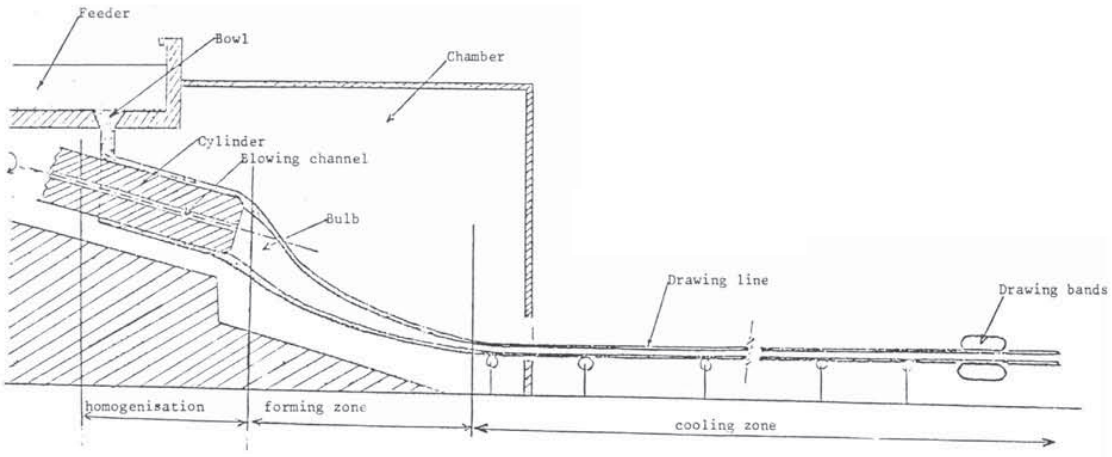


Figure 4-3: Glass Tube Manufacturing Process [68]. This article was published in the 10th Triennial IFAC Congress on Automatic Control, Volume 20, V. Wertz and G. Bastin and M. Haest, Identification of a glass tube drawing bench, Page 334, © Elsevier (1987).

Modeling and Physical Insight

The two main controlled inputs to the process are the drawing speed and the blowing pressure. The outputs are the diameter and thickness of the manufactured glass tube. In this work, the objective is to identify the process between the drawing speed and measured thickness. Modeling this relationship can be a challenging task. The paper presenting the process concluded little effect of pressure on the thickness of the final glass tube by using correlation analysis. In addition, the model used to identify this dynamic relationship is linear model with 4 samples of delay [68].

Benchmark Data Description

The output contains high frequency components that proved difficult to deal with in the tutorial [20]. The datasets are hence decimated by 4. The signals are also detrended to bring means to zero and get rid of the static term. The final pre-processed signals used for system identification are shown in figures 4-4 and 4-5. The latter contains 625 samples with a sampling time of 4 seconds. The first 500 constitute the estimation signal and the remaining 125 samples the validation signal.

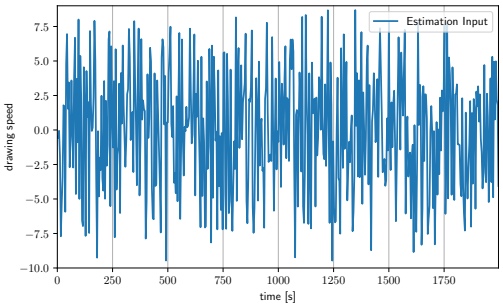


Figure 4-4: Glass Tube Manufacturing estimation input data.

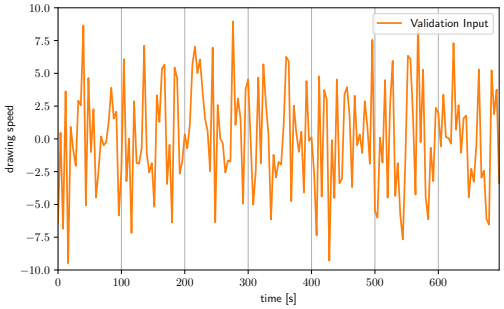


Figure 4-5: Glass Tube Manufacturing validation input data.

Identification Challenges

The temperature in the bowl before the feeder and the perturbation in the cylinder rotations are both reported to be sources of disturbances to the process under study [68]. Furthermore, a delay of four samples (for the pre-processed data) exist. This is caused by a more distant placement of the sensor from the chamber's output.

Previous Identification Works

Table 4-2: Comparison of other models from literature on Glass Tube Manufacturing Dataset

Previous Work	Method	Tests	RMSE
[20] The Mathworks Incorporation	ARX Model Estimation `arx(data, [1, 1, 3])`	One-Step Ahead Prediction	0.649
		Free Run Simulation	0.688
[20] The Mathworks Incorporation	Subspace Identification `n4sid(data)`	One-Step Ahead Prediction	0.784
		Free Run Simulation	0.676

4-4-3 Hair Dryer

The laboratory hair dryer is a system that refers to the PT 326 Process Trainer. A schematic of the system is shown in figure 4-6. In this process, air is blown from the surrounding into a tube with a centrifugal fan and heated using a grid of resistors wires. The outlet air temperature is measured by a bead thermistor. Both the air flow and the voltage provided on the resistor wires are inputs to the Hair Dryer. The data can be loaded on Matlab using the command `load dryer2` [18].

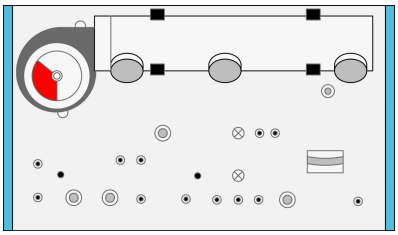


Figure 4-6: PT 326 Process Trainer Sketch

Modeling and Physical Insight

A attempt to model the process to be identified, the balance of energy equation is used in a similar fashion to the previous derivation for the heat exchanger. Assuming the flow incompressible and the effect of ambient temperature is negligible, the accumulated heat in the hair dryer is given by:

$$\dot{E}_{acc} = \dot{E}_{in} - \dot{E}_{out} \tag{4-3}$$

$$mc\,\frac{d(\Delta T)}{dt} = \dot{m}c\,\Delta T + \dot{E}_{\Omega} \tag{4-4}$$

With $\Delta T = T_{in} - T_{out}$. c represent the air heat capacity of respectively the coolant and the product fluid. m is the fluid mass and $\dot{m}$ the mass flow rate. The heat flux from the resistor mesh is assumed proportional to the input voltage $\dot{E}_{\Omega} = KV_i$. This results in a linear model relating the change in temperature and the input voltage.

Benchmark Data Description

The tutorial "Estimating Simple Models from Real Laboratory Process Data" uses a binary random input voltage of a 1000 datapoints. The signals' sampling time is 0.08 seconds [18]. The first 300 samples are used for estimation and 700 samples for validation. Plots of the used inputs are given in figures 4-7 and 4-8.

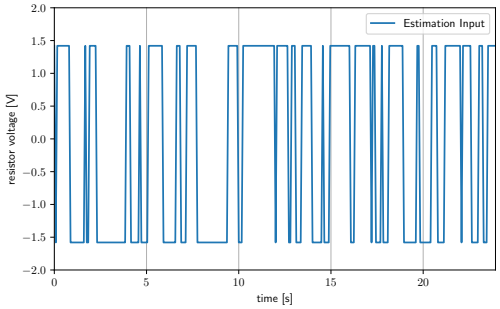


Figure 4-7: Hair Dryer estimation input data.

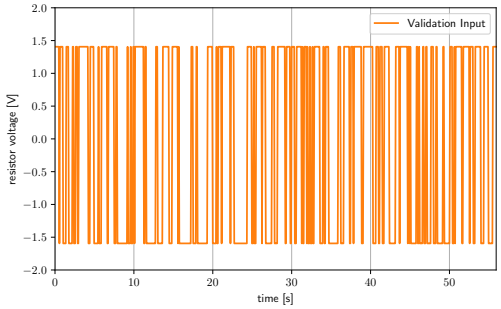


Figure 4-8: Hair Dryer validation input data.

Identification Challenges

The system has little disturbances coming from the ambient temperature and the signals exhibit a good signal to noise ratio [31]. Only a delay of 3 samples is reported in the tutorial [18].

Previous Identification Works

In the Matlab tutorial the first 300 datapoints are used for estimation and a 100 of the remaining samples for validation [18]. However, in Lennart Ljung's Book the dataset was divided in half [31]. In this work, both works will be mentioned as previous works. The model structures and procedures will be kept as the authors determined it but the estimation and validation data will correspond to those used for the estimation of this work's model. In other words, the Matlab and book methods used to identify this same process are rerun to obtain a compatible validation comparison using Matlab.

Table 4-3: Comparison of other models from literature on Hair Dryer Dataset

Previous Work	**Method**	**Tests**	**RMSE** $[V]$
[18] The Mathworks Incorporation	Transfer Function Estimation `tfest(data, 2, 1)`	One-Step Ahead Prediction	0.085
		Free Run Simulation	0.108
[18] The Mathworks Incorporation / Ljung's Book Chapter 17.3 [31]	Subspace Identification `n4sid(data, 3)`	One-Step Ahead Prediction	0.037
		Free Run Simulation	0.106
[31] Ljung's Book Chapter 17.3	Subspace Identification `n4sid(data, 6)`	One-Step Ahead Prediction	0.036
		Free Run Simulation	0.161
[18] The Mathworks Incorporation / Ljung's Book Chapter 17.3 [31]	ARX Model Estimation `arx(data, [2, 2, 3])`	One-Step Ahead Prediction	0.039
		Free Run Simulation	0.106
[31] Ljung's Book Chapter 17.3	ARX Model Estimation `arx(data, [6, 9, 2])`	One-Step Ahead Prediction	0.036
		Free Run Simulation	0.103
[31] Ljung's Book Chapter 17.3	ARMAX Model Estimation `armax(data, [3, 3, 2, 2])`	One-Step Ahead Prediction	0.036
		Free Run Simulation	0.104

4-4-4 Cascaded Tanks

The cascaded tanks are 2 water tanks placed on top of each others over a reservoir. Water in these can flow in the direction of gravity except for the bottom reservoir. A pump is used to pump water into the top tank. What is particular about this setup is the fact that overflow is possible from the tanks to the reservoir. That saturation behavior is an often encountered non-linearity in non-linear systems. The input of the process to be identified is the pump voltage and the output is the water level of the second tank measured using a capacitive level sensor.

Modeling and Physical Insight

A sketch of the system in study is given in figure 4-9. x_1 is the height of fluid in the upper tank and x_2 the height in the bottom tank. The area of the orifice at the bottom of the upper tank is given by a_1 and the one at the bottom of the lower tank is a_2. A_1 is the cross-sectional area of the upper tank and A_2 the cross sectional area of the bottom tank. The pump constant is denoted as K_p and the pump flow rate is assumed linear with the input voltage u, $\dot{m}_p = K_p u$.

Finally the mass flow rate out of the upper tank is designated by $\dot{m}_1^{out}$ and out of the lower tank is $\dot{m}_2^{out}$. By effect of mass conservation in the control volume of each of the tanks,

$$A_1 \frac{d(x_1)}{dt} = \dot{m}_p - \dot{m}_{out}^1 \tag{4-5}$$

$$A_2 \frac{d(x_2)}{dt} = \dot{m}_{out}^1 - \dot{m}_{out}^2 \tag{4-6}$$

Using Bernoulli's law considering static pressure atmospheric and flow incompressible, the output mass flow rates from each tank are given by

$$\dot{m}_{out}^1 = a_1 v_1^{out} = a_1 \sqrt{2gx_1} \tag{4-7}$$

$$\dot{m}_{out}^2 = a_2 v_2^{out} = a_2 \sqrt{2gx_2} \tag{4-8}$$

where g is the gravity constant.

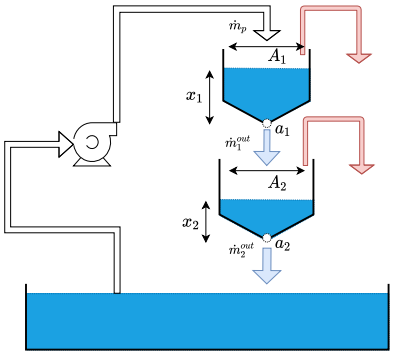


Figure 4-9: Sketch of the Cascaded Tanks system

Using these expressions in equations 4-5 and 4-6, one obtains the differential equations describing the process

$$\frac{d(x_1)}{dt} = \frac{K_p}{A_1} u - \frac{a_1\sqrt{2g}}{A_1}\sqrt{x_1} + \omega_1 \tag{4-9}$$

$$\frac{d(x_2)}{dt} = \frac{a_1\sqrt{2g}}{A_2}\sqrt{x_1} - \frac{a_2\sqrt{2g}}{A_2}\sqrt{x_2} + \omega_2 \tag{4-10}$$

$$y = x_2 + \epsilon \tag{4-11}$$

with ω_1 and ω_2 the process noise and ϵ the measurement noise.

Benchmark Data Description

Two multisine input benchmark signals (0-0.0144 Hz) are provided for the identification of the Cascaded Tanks system. These contain each 1024 samples with a sampling time of 4 seconds. The system state have a similar unknown initial value for both estimation and validation data [57]. To bring the static bias term to zero the datasets are detrended before model estimation.

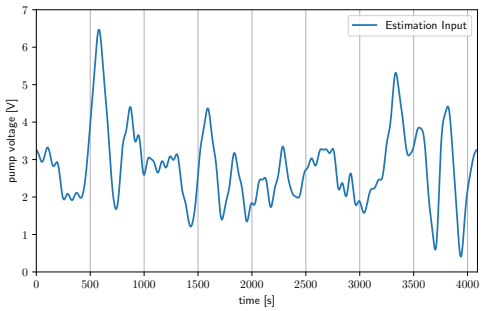


Figure 4-10: Cascaded Tanks estimation input data.

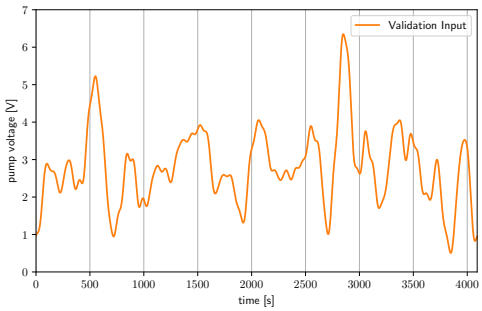


Figure 4-11: Cascaded Tanks validation input data.

Identification Challenges

In addition to the non-linear nature of the Cascaded Tanks system, saturation introduces input-dependent process noise because of overflow from first to second tank [57].

Previous Identification Works

Table 4-4: Comparison of other models from literature for Cascaded Tanks Benchmark

Paper	Method	Test Used	Measure	RMSE [V]
[3] Belz et al. (2017)	Local model Networks with NFIR for external dynamics	One-Step Ahead Prediction	RMSE	0.0573
		Free Run Simulation	RMSE	0.669
[24] Karagoz et al. (2020)	Regularized tensor network B-splines	One-Step Ahead Prediction	RMSE	0.0461
		Free Run Simulation	RMSE	0.3018
[59] Schoukens et al. (2016)	Best Linear Approximation	One-Step Ahead Prediction	RMSE	0.0556
		Free Run Simulation	RMSE	0.5878
[59] Schoukens et al. (2016)	Polynomial Feedback Model	One-Step Ahead Prediction	RMSE	0.0555
		Free Run Simulation	RMSE	0.4877
[59] Schoukens et al. (2016)	Volterra Feedback Model	One-Step Ahead Prediction	RMSE	0.0494
		Free Run Simulation	RMSE	0.3972
[16] Hostettler et al. (2018)	GP Drift Model	One-Step Ahead Prediction	RMSE	0.0576
[59] Schoukens et al. (2016)	Polynomial Hammerstein Model	Free Run Simulation	RMSE	0.5651
[59] Schoukens et al. (2016)	Polynomial Wiener Model	Free Run Simulation	RMSE	0.5086
[50] Relan et al. (2017)	Non-Linear State Space Model	Free Run Simulation	RMSE	0.3433
[50] Relan et al. (2017)	Polynomial Non-Linear State Space Model	Free Run Simulation	RMSE	0.44984
[63] Svensson et al. (2017)	Flexible State Space Model	Free Run Simulation	RMSE	0.45
[35] Mattsson et al. (2018)	Piecewise ARX Models	Free Run Simulation	RMSE	0.350
[6] Brunot et al. (2017)	OEM with Nelder Mead Solver	Free Run Simulation	RMSE	0.379
[6] Brunot et al. (2017)	OEM with NOMAD Solver	Free Run Simulation	RMSE	0.376

4-4-5 Coupled Electric Drives

Coupled Electric Drives is mechanical system that consists of an elastic belt connecting two motors and a pulley. The pulley is attached to a suspended spring. Motors can be used to allow a control of both the tension and the speed of the belt. The provided data is measured speed by a pulse counter on the pulley, making it insensitive to the sign of velocity.

Modeling and Physical Insight

The system's dynamics can be hard to derive using physical laws and may need multiple assumptions over the forces and coupling caused by the flexible belt. For a detailed modeling of this system please refer to the derivation by Petr et al. [9]. Nonetheless, five main modes are perceived in CED: the two electric drives time constant, the spring and the analogue low-pass filter [69] and the pulley inertial forces. Other energy storage can include the flexible band elastic forces. In the technical note provided along the dataset, a Wiener/Wiener-Hammerstein model is assumed and its parameters are identified [69]. Furthermore the sensor output is $y(t) = |z(t)| + e(t)$. e represents the output disturbance and $|z(t)|$ is the absolute value of the dynamic model output given the nature of the sensor measurement.

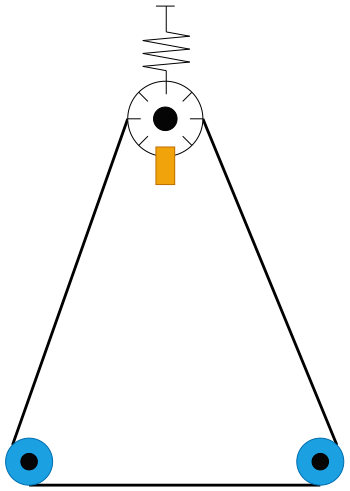


Figure 4-12: Coupled Electric Drive Sketch

Benchmark Data Description

Two sets of uniform pulses with various random amplitudes inputs are provided and are used for training and validation. They are divided into 300 samples for estimation and 200 for validation each. These are shown in figures 4-13 4-14 below.

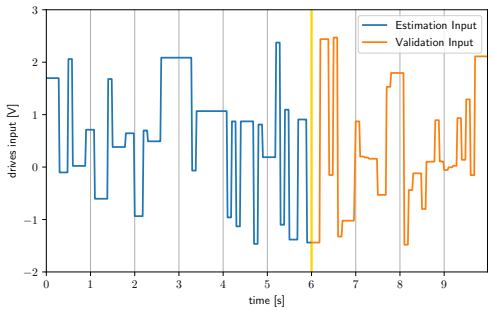


Figure 4-13: First CED provided input data.

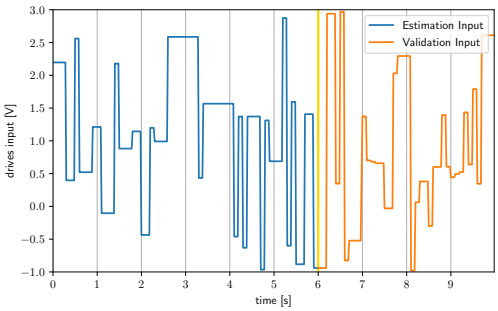


Figure 4-14: Second CED provided input data.

Identification Challenges

The CED exhibits a strong non-linearity given the absolute value in the measurement of the velocity. Adding to that, estimation datasets are short [69] and physical modeling can prove to be a tedious task because any assumption on the coupling caused by the elastic belt can cause an under specification of the true mechanism.

Previous Identification Works

Table 4-5: Comparison of other models from literature for CED Benchmark

Paper	**Method**	**Test Used**	**Measure**	**RMSE [ticks/s]**	
				UNIF 1	UNIF 2
[1] Ayala et al.	Cascaded Evolutionary Algorithm using RBF Neural Networks and Lipschitz Regressors [14]	One-Step Ahead Prediction	MSE	0.0592	0.0678
		Free Run Simulation	MSE	0.435	0.833
[1] Ayala et al.	Cascaded Evolutionary Algorithm using RBF Neural Networks with Free Search Differential Evolution regressors	One-Step Ahead Prediction	MSE	0.0412	0.04
		Free Run Simulation	MSE	0.130	0.185
[39] Nechita et al.	Tree Adjoining Grammars	One-Step Ahead Prediction	RMSE	-	0.032
		Free Run Simulation	RMSE	-	0.128
[40] Nejib et al.	Support-Vector Regression with RBF Kernel.	One-Step Ahead Prediction	MNSE	0.0305	-
[52] Sabahi et al.	Fuzzy System Identification.	Free Run Simulation	RMSE	0.5607	0.3228
[52] Sabahi et al.	Extended Fuzzy Logic (FLe) System for System Identification.	Free Run Simulation	RMSE	0.15044	0.09209
[54] Scarpinitti et al.	Cascaded Spline Adaptive NonLinear Identification.	Free Run Simulation	MSE_{dB}	0.2165	0.110

4-4-6 Bouc-Wen Hysteresis Model

Hysteresis is a lagging phenomenon caused by a dependency on precedent states and inputs. Hysteric effects exist in both social and physical sciences. Some examples include unemployment in economics, amplifier circuits in electrical systems, and random vibrations or magnetization in mechanical systems. Modeling such systems using physical laws is an arduous task [22]. The Bouc-Wen model is a semi-physical model, combining both physical understanding and black box modeling. It is a dynamic non-linear, single degree of freedom system relating displacement and hysteric restoring force [22].

Modeling and Physical Insight

In the last decade, a lot of attention has been given to the Bouc-Wen model, and tuning its parameters for hysteresis applications [22]. With a single mass m_L, governed by Newton's law, the differential equation describing the system is written as [58]:

$$m_L\ddot{y}(t) + r(y,\dot{y}) + z(y,\dot{y}) = u(t) \tag{4-12}$$

where , y is the output displacement and u the external force. $r(y,\dot{y})$ is the static restoring force, and $z(y,\dot{y})$ the hysteric force. Both these components obey the following set of equations:

$$r(y,\dot{y}) = k_L y + c_L \dot{y} \tag{4-13}$$

$$\dot{z}(y,\dot{y}) = \alpha\dot{y} - \beta(\gamma|\dot{y}||z|^{\nu-1}z + \delta\dot{y}|z|^{\nu}) \tag{4-14}$$

k_L and c_L being respectively the linear stiffness and viscous damping factors. α, β, γ, δ, and ν are parameters tuned to shape specific hysteric application loops.

Benchmark Data Description

The benchmark data provides fixed datasets for validation, one is a random multisine and the other a sinesweep signal. For estimation, a Matlab Simulink package provides a simulation of the system, through Newmark integration of the dynamics [58]. In this thesis, and referring to previous works on this benchmark and recommendations of the package owners, a random multisine is used with 5-150 Hz excited frequencies and a RMS amplitude of 50 N for estimation. Finally, the sampling frequency is set to 750 Hz. The data inputs used for system identification and validation are shown in figures 4-15, 4-16 and 4-17 below:

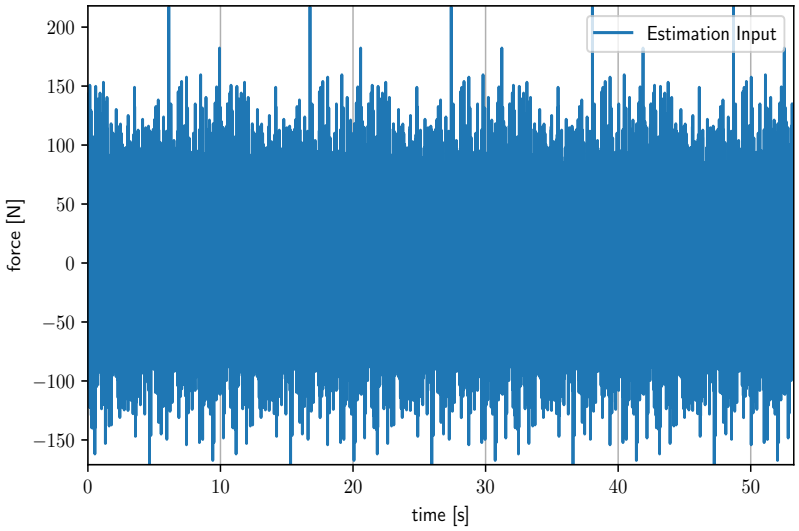


Figure 4-15: Bouc-Wen estimation input data.

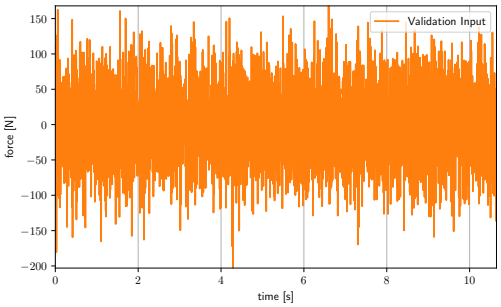


Figure 4-16: Multisine Validation Input for BW Benchmark.

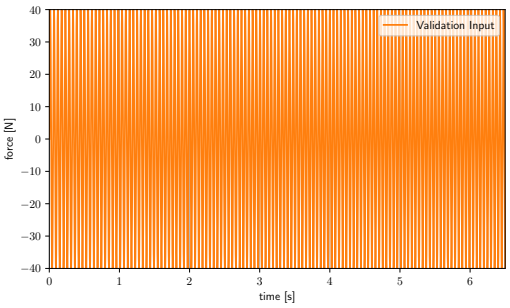


Figure 4-17: Sinesweep Validation Input for BW Benchmark.

A band limited Gaussian noise (0-375 Hz) is added to the output with RMS of 0.008 mm for the estimation data only. The estimation and validation data have different scales, the output RMS is around $6.65 \cdot 10^{-4}$. Thus, the estimation and validation datasets are normalized.

Identification Challenges

The main identification challenge of this benchmark is the dynamic non-linearity shown in equation 4-14. This non-linearity is governed by the non-measurable internal state z.

Previous Identification Works

Table 4-6: Comparison of other models from literature for Bouc-Wen Benchmark

Paper	**Method**	**Test Used**	**Measure**	**RMSE [mm] ($\cdot 10^{-4}$)**	
				Multisine	Sinesweep
[3] Belz et al. (2017)	Automatic Model Generation with Local model Networks (NARX/NFIR)	One-Step Ahead Prediction	RMSE	0.0986	0.0687
		Free Run Simulation	RMSE	1.6356	1.380
[39] Nechita et al.	Tree Adjoining Grammars	One-Step Ahead Prediction	RMSE	-	0.0737
		Free Run Simulation	RMSE	-	0.652
[59] Schoukens et al. (2016)	Best Linear Approximation	One-Step Ahead Prediction	RMSE	0.1126	0.0698
		Free Run Simulation	RMSE	1.5105	1.6619
[59] Schoukens et al. (2016)	Polynomial Feedback Model	One-Step Ahead Prediction	RMSE	0.0195	0.0451
		Free Run Simulation	RMSE	1.2091	1.5004
[59] Schoukens et al. (2016)	Volterra Feedback Model	One-Step Ahead Prediction	RMSE	0.0895	0.0347
		Free Run Simulation	RMSE	0.8755	0.6392
[16] Hostettler et al. (2018)	GP Drift Model	One-Step Ahead Prediction	RMSE	0.0580	0.0096
[59] Schoukens et al. (2016)	Polynomial Hammerstein Model	Free Run Simulation	RMSE	1.4967	1.8691
[59] Schoukens et al. (2016)	Polynomial Wiener Model	Free Run Simulation	RMSE	1.4877	1.6235
[6] Brunot et al. (2017)	OEM parametric estimation based on Nelder Mead Solver	Free Run Simulation	RMSE	0.468	0.0186
[6] Brunot et al. (2017)	OEM parametric estimation based on NOMAD Solver	Free Run Simulation	RMSE	0.468	0.0190
[10] Fakhrizadeh et al. (2017)	Polynomial State-Space Model	Free Run Simulation	RMSE	0.1870	0.1202