Neural State-Space Models: Empirical Evaluation of Uncertainty Quantification

Marco Forgione[1] and Dario Piga[1]

[1]IDSIA Dalle Molle Institute for Artificial Intelligence USI-SUPSI, Lugano, Switzerland. (name.surname@supsi.ch).

April 3, 2023

Abstract

Effective quantification of uncertainty is an essential and still missing step towards a greater adoption of deep-learning approaches in different applications, including mission-critical ones. In particular, investigations on the predictive uncertainty of deep-learning models describing non-linear dynamical systems are very limited to date. This paper is aimed at filling this gap and presents preliminary results on uncertainty quantification for system identification with neural state-space models. We frame the learning problem in a Bayesian probabilistic setting and obtain posterior distributions for the neural network's weights and outputs through approximate inference techniques. Based on the posterior, we construct credible intervals on the outputs and define a *surprise index* which can effectively diagnose usage of the model in a potentially dangerous out-of-distribution regime, where predictions cannot be trusted.

Please cite this version of the paper:

M. Forgione and D. Piga. Neural State-Space Models: Empirical Evaluation of Uncertainty Quantification. In *Proc. of the 22nd IFAC World Congress, Yokohama, Japan*, 2023.

You may use the following bibtex entry:

```
@inproceedings{forgione2023empirical,
  title={{Neural State-Space Models: Empirical Evaluation
          of Uncertainty Quantification}},
  author={Forgione, Marco and Piga, Dario},
  booktitle={Proc. of the 22nd IFAC World Congress, Yokohama, Japan},
  year={2023}
}
```

1 Introduction

In recent years, the system identification community has shown renewed interest in deep-learning tools and techniques for data-driven modeling of non-linear dynamical systems (Ljung et al., 2020). To cite a few examples, system identification approaches based on 1-D convolutional neural networks are presented in Andersson et al. (2019); Wu and Jahanshahi (2019). Training of neural NARX architectures with a regularization term promoting decay of the model's linearized impulse response are introduced in Peeters et al. (2022). Neural networks architectures and fitting criteria for continuous-time dynamical model identification are presented in Mavkov et al. (2020). Finally, algorithms for efficient training of tailor-made neural state-space models are discussed in Forgione and Piga (2020) and Beintema et al. (2021).

A common (and justified) criticism to the above-mentioned deep system identification approaches is the general lack of uncertainty description and analysis. Indeed, the methods presented in those contributions only produce *nominal* point predictions, with no explicit measure of their reliability. While the models are shown to deliver high performance in the considered benchmarks, results may dramatically deteriorate when they are used in an *out-of-distribution* regime, i.e. on a test set whose characteristics (in terms of input amplitude, frequency, power, etc.) differ significantly from the ones of the training data. Even worse, no mechanism is in place to *detect* this failure mode, and models may quietly produce off-target, possibly dangerous predictions.

In current machine learning research, uncertainty quantification is recognized as paramount to increase reliability and acceptance of black-box models like neural networks, and it is thus seen as a fundamental step towards their adoption in mission-critical applications (Loquercio et al., 2020). Different approaches, both deterministic and probabilistic, have been proposed, see Gawlikowski et al. (2021) for a recent survey. The probabilistic perspective is arguably more general and theoretically appealing. Certain methodologies like ensemble learning (Lakshminarayanan et al., 2017) and dropout (Srivastava et al., 2014), first introduced in a deterministic settings, are now better understood as approximate inference algorithms in a Bayesian probabilistic framework.

Most of the contributions on uncertainty quantification presented in the deep-learning literature involve static regression problems (typically from the UCI datasets) with feed-forward neural architectures and/or image classification problems (typically from the CIFAR dataset and variants thereof) with convolutional ones, see Maddox et al. (2019); Wilson and Izmailov (2020); Izmailov et al. (2021). To date, little attention has been devoted to sequential learning problems and in particular to non-linear dynamical systems modeling.

A notable exception is the recent contribution (Zhou et al., 2022), where learning of dynamical systems in neural input/output form is formulated in a Bayesian probabilistic framework. Compared to (Zhou et al., 2022), our work is focused on neural *state-space* models, which are arguably more suitable for downstream control applications (e.g. for model predictive control) and for analysis with standard system theoretic tools. Furthermore, the main objectives in (Zhou et al., 2022) are to select the relevant input regressors and to induce sparsity in the network, while our work is focused on uncertainty description and recognition of the out-of-distribution regime.

We obtain uncertainty bounds by framing the neural state-space identification problem in a Bayesian probabilistic settings and by deriving (approximate) posterior distributions for the neural network parameters and for its output predictions. From a technical perspective, we use the Laplace approximation (Bishop and Nasrabadi, 2006) to describe the parameter posterior distribution and exploit our recent results in (Forgione et al., 2022) to speed up the required Hessian matrix computations. We show that the obtained uncertainty bounds, while not always calibrated, widen significantly when neural state-space models are used in an out-of-distribution regime.

Based on the obtained uncertainty description, we then introduce a new metric, called *surprise* index that, for a given trained model and a new input sequence, detects whether the model is suitable to predict the corresponding output before collecting any new data. Thus, the surprise index may be used to assess beforehand whether the predictions generated by a model fed by a specific input signal can be trusted.

We demonstrate the effectiveness of our methodology on a variation of the Wiener-Hammerstein identification benchmark (Schoukens et al., 2009) conveniently modified to generate data from different regimes, and release the codes required to reproduce our results in the GitHub repository `https://github.com/forgi86/sysid-neural-unc`.

2 Methodology

2.1 Dataset and objective

We are given a dataset $\mathcal{D} = (\mathbf{u}, \mathbf{y})$ with N input samples $u_k \in \mathbb{R}^{n_u}$ and (possibly noisy) output samples $y_k \in \mathbb{R}$, collected from a dynamical data-generating system $\mathcal{S}$.

Our goal is to estimate from $\mathcal{D}$ a neural state-space model M of the unknown dynamics of $\mathcal{S}$, which for a new input sequence $\mathbf{u}^*$, generates a prediction of the corresponding output $\mathbf{y}^*$, plus an *indicator* of the predictions' reliability.

Given a suitable probabilistic neural model structure with a prior distribution defined over its parameters, the problem may be tackled through (approximate) statistical inference of the *posterior predictive distribution* (ppd) $p(\mathbf{y}^*|\mathbf{u}^*, \mathcal{D})$, which in turn may be used to generate output predictions with credible intervals.

In this paper, we follow indeed a probabilistic approach bearing in mind that, due to assumptions and approximations introduced to carry out the inference step efficiently, the obtained ppd and bounds may be somewhat inaccurate. Still, we aim at exploiting probabilistic reasoning and tools to obtain useful indicators of model predictions' reliability. These indicators should (at least) be able to detect when the model is operating in an extrapolation regime, and thus its prediction cannot be fully trusted.

The case of multi-input single-output systems is discussed to simply exposition. However, the results can be extended straightforwardly to multi-input multi-output systems.

2.2 Model structure

We consider the following neural state-space model structure M:

$$\begin{aligned} x_{k+1} &= \mathcal{F}(x_k, u_k; \theta) && \text{(1a)}\\ \hat{y}_k &= \mathcal{G}(x_k; \theta) && \text{(1b)}\\ y_k &= \hat{y}_k + e_k, \qquad e_k \sim \mathcal{N}\left(0, {}^1\!/\!_\beta\right) && \text{(1c)}\\ \theta &\sim \mathcal{N}\left(0, {}^1\!/\!_\tau\right), && \text{(1d)} \end{aligned}$$

where $\mathcal{F}$ and $\mathcal{G}$ are feed-forward neural networks having compatible dimensions, $x_k \in \mathbb{R}^{n_x}$ is the state at time k, and $\theta \in \mathbb{R}^{n_\theta}$ is a vector of parameters to be estimated from data. The measured output $y_k \in \mathbb{R}$ is assumed to be corrupted by a zero-mean white Gaussian noise error e with precision β. The prior on the model parameters θ is also Gaussian, with precision τ.

3 Probabilistic derivations

3.1 Posterior parameter distribution

The posterior distribution $p(\theta|\mathcal{D})$ of θ conditioned on the observations $\mathcal{D}$ is given by the Bayes rule:

$$p(\theta|\mathcal{D}) = \frac{p(\theta)p(\mathcal{D}|\theta)}{p(\mathcal{D})}. \tag{2}$$

The functional form of the Gaussian *prior* distribution $p(\theta)$ on the model parameters θ is:

$$p(\theta) = \frac{\tau^{n_\theta}}{\sqrt{(2\pi)^{n_\theta}}} \exp\left(\frac{-\tau}{2} \sum_{i=0}^{n_\theta - 1} \theta_i^2\right), \tag{3}$$

while the *likelihood* $p(\mathcal{D}|\theta)$ is:

$$p(\mathcal{D}|\theta) = \frac{\beta^N}{\sqrt{(2\pi)^N}} \exp\left(\frac{-\beta}{2} \sum_{k=0}^{N-1} (y_k - \hat{y}_k(\theta))^2\right), \tag{4}$$

3.2 Posterior predictive distribution

The posterior predictive distribution $p(\mathbf{y}^*|\mathbf{u}^*, \mathcal{D})$ given a new input sequence $\mathbf{u}^*$ is:

$$p(\mathbf{y}^*|\mathbf{u}^*, \mathcal{D}) = \int_\theta p(\mathbf{y}^*|\mathbf{u}^*, \theta)p(\theta|\mathcal{D})\; d\theta. \tag{5}$$

Even when the approximate $p(\theta|\mathcal{D})$ has a simple structure, exact solution of the integral above is intractable and further approximations/simplifications are required to evaluate the ppd.

4 Approximate inference

We present in this section the approximate inference approaches used to obtain the parameter posterior $p(\theta|\mathcal{D})$ and the predictive posterior $p(\mathbf{y}^*|\mathbf{u}^*, \mathcal{D})$.

4.1 Laplace approximation of the parameter posterior

The parameter posterior $p(\theta|\mathcal{D})$ is approximated using the Laplace's method (Bishop and Nasrabadi, 2006) centered around the maximum a posteriori (MAP) point estimate θ^{MAP}. All the derivations and required computations are specified in this section.

4.1.1 MAP point estimate

To obtain the MAP estimate, we consider the *negative* logarithm of the posterior distribution $\mathcal{L}(\theta) = -\log p(\theta|\mathcal{D})$:

$$\mathcal{L}(\theta) = \overbrace{\frac{\beta}{2}\sum_{k=0}^{N-1}(y_k - \hat{y}_k(\theta))^2}^{=E_{\mathrm{lik}}(\theta)} + \overbrace{\frac{\tau}{2}\sum_{i=0}^{n_\theta-1}\theta_i^2}^{=E_{\mathrm{prio}}(\theta)} + \mathrm{cnst}, \tag{6}$$

where cnst is a term that does not depend on θ.

The MAP estimate θ^{MAP} is:

$$\theta^{\mathrm{MAP}} = \arg\min_{\theta} \mathcal{L}(\theta). \tag{7}$$

Computation of θ^{MAP} corresponds to a non-linear (regularized) least-squares problem, which for neural state-space models is usually tackled with stochastic gradient descent algorithms or variants thereof.

4.1.2 Laplace approximation

The Laplace approximation of the parameter posterior distribution centered around the MAP estimate is defined as:

$$p(\theta|\mathcal{D}) = \mathcal{N}(\theta^{\mathrm{MAP}}, P_{\theta^{\mathrm{MAP}}}), \tag{8}$$

where $P_{\theta^{\mathrm{MAP}}}$ is the inverse of the Hessian of the negative log-likelihood evaluated in θ^{MAP}:

$$P^{-1}_{\theta^{\mathrm{MAP}}} = \left.\frac{\partial^2 \mathcal{L}(\theta)}{\partial \theta^2}\right|_{\theta=\theta^{\mathrm{MAP}}}. \tag{9}$$

The Hessian of $E_{\text{prio}}(\theta)$ has the simple functional form:

$$\frac{\partial^2 E_{\text{prio}}(\theta)}{\partial \theta^2} = \tau I, \tag{10}$$

while the Hessian of $E_{\text{lik}}(\theta)$ is:

$$\frac{\partial^2 E_{\text{lik}}(\theta)}{\partial \theta^2} = \beta \sum_{k=0}^{N-1} \frac{\partial \hat{y}_k}{\partial \theta} \frac{\partial \hat{y}_k}{\partial \theta}^\top + \beta \sum_{k=0}^{N-1} (\hat{y}_k - y_k) \frac{\partial^2 \hat{y}_k}{\partial \theta^2}. \tag{11}$$

According to the Gauss-Newton (GN) Hessian approximation (Wright et al., 1999), the expression above is dominated by the first term $\beta \sum_{k=0}^{N-1} \frac{\partial \hat{y}_k}{\partial \theta} \frac{\partial \hat{y}_k}{\partial \theta}^\top$ and the second contribution $\beta \sum_{k=0}^{N-1} (\hat{y}_k - y_k) \frac{\partial^2 \hat{y}_k}{\partial \theta^2}$ may be neglected. The GN approximation is accurate, for instance, when $\frac{\partial^2 \hat{y}_k}{\partial \theta^2}$ is small (i.e. model predictions are nearly affine), when $\hat{y}_k - y_k$ is small (which is typically the case for the optimal θ if the variance of e_k is also small) and, more in general, when $\frac{\partial^2 \hat{y}_k}{\partial \theta^2}$ and $\hat{y}_k - y_k$ are uncorrelated (which is also expected for the optimized value of θ, as the residual $\hat{y}_k - y_k$ is then close to the white measurement noise e_k).

Overall, the covariance matrix $P_{\theta^{\text{MAP}}}$ with GN Hessian approximation is:

$$P_{\theta^{\text{MAP}}}^{-1} \approx \tau I + \beta \sum_{k=0}^{N-1} \frac{\partial \hat{y}_k}{\partial \theta} \frac{\partial \hat{y}_k}{\partial \theta}^\top. \tag{12}$$

Remark 1 *The term* $\beta \sum_{k=0}^{N-1} \frac{\partial \hat{y}_k}{\partial \theta} \frac{\partial \hat{y}_k}{\partial \theta}^\top$ *in* (12) *is also a finite-sample approximation of the* Fisher Information Matrix*, which corresponds in frequentist statistics to the (asymptotic) precision of the maximum likelihood estimator (Van den Bos, 2007). In this sense, the methodologies of this paper are also applicable to the derivation of confidence intervals in a frequentist setting.*

4.1.3 Computational aspects

A straightforward approach to obtain of the gradients $\frac{\partial \hat{y}_k}{\partial \theta}$, $k = 0, \dots, N-1$ needed in (11) is to invoke N independent back-propagation operations through the neural state-space model's unrolled computational graph at each time step. Overall, this naive approach requires a number of operations $\mathcal{O}(N^2)$.

The computational cost can actually be lowered to $\mathcal{O}(N)$ using the recursive methodology based on sensitivity equations first introduced by the authors in Forgione et al. (2022) and reported hereafter for completeness.

Let us introduce the *state sensitivities* $s_k = \frac{\partial x_k}{\partial \theta} \in R^{n_x \times n_\theta}$. By taking the derivatives of the left- and right-hand side of (1a) w.r.t. the model parameters θ, we obtain a recursive equation describing the evolution of s_k:

$$s_{k+1} = J_k^{fx} s_k + J_k^{f\theta}, \tag{13}$$

where $J_k^{fx} \in \mathbb{R}^{n_x \times n_x}$ and $J_k^{f\theta} \in \mathbb{R}^{n_x \times n_\theta}$ are the Jacobians of $\mathcal{F}(x_k, u_k; \theta)$ w.r.t. x_k and θ, respectively.

Let us now take the derivative of (1b) w.r.t. θ:

$$\frac{\partial \hat{y}_k}{\partial \theta} = J_k^{gx} s_k + J_k^{g\theta}, \tag{14}$$

where $J_k^{gx} \in \mathbb{R}^{n_y \times n_x}$ and $J_k^{g\theta} \in \mathbb{R}^{ny \times n_\theta}$ are the Jacobians of $\mathcal{G}(x_k, u_k; \theta)$ w.r.t. x_k and θ, respectively, and n_y is the number of outputs ($n_y = 1$ in this paper).

The Jacobians J_k^{fx} and $J_k^{f\theta}$ can be obtained through n_x back-propagation operations through $\mathcal{F}$, thus at cost $\mathcal{O}(n_x n_\theta)$. Similarly, J_k^{gx} and $J_k^{g\theta}$ can be obtained through n_y back-propagation operations through $\mathcal{G}$ at cost $n_y n_\theta$. Thus, the computational effort required to obtain $\frac{\partial \hat{y}_k}{\partial \theta}$ in (14) (given the previous sensitivity s_{k-1}) is $\mathcal{O}((n_x + n_y)n_\theta)$. Overall, all the derivatives of interest $\frac{\partial \hat{y}_k}{\partial \theta}$, $k = 0, \dots, N-1$ are then computed at a total cost $\mathcal{O}(N(n_x + n_y)n_\theta)$.

4.2 Linearization-based approximation of the ppd

Our approximation of the posterior predictive distribution is based on a linearization of the neural network model with respect to its parameters about the MAP estimate:

$$\hat{\mathbf{y}}^*(\theta) \approx \hat{\mathbf{y}}^*(\theta^{\mathrm{MAP}}) + J^*(\theta - \theta^{\mathrm{MAP}}), \tag{15}$$

where J^* is the Jacobian of $\hat{\mathbf{y}}^*$ with respect to the parameters θ, computed for $\theta = \theta^{\mathrm{MAP}}$. According to the approximation above, we obtain:

$$\mathbf{y}^* \sim \mathcal{N}\left(\hat{\mathbf{y}}^*(\theta^{\mathrm{MAP}}),\ \overbrace{J^* P_{\theta^{\mathrm{MAP}}} J^{*\top} + \frac{1}{\beta} I}^{=\Sigma_{\mathbf{y}^*}}\right). \tag{16}$$

Note that the ppd's covariance matrix is the sum of a term $J^* P_{\theta^{\mathrm{MAP}}} J^{*\top}$ related to the approximate knowledge of the true system parameters (our epistemic uncertainty), plus a term $^1/_\beta I$ related to measurement noise (the intrinsic aleatoric uncertainty).

We are interested in particular into the diagonal entries of $\Sigma_{\mathbf{y}^*}$, which correspond to the variance of the output predictions at the different time steps and thus represent their uncertainty. Specifically, we construct and visualize 99.7% credible intervals centered around the nominal prediction and having width ±3 times the square root of these diagonal entries.

4.2.1 Surprise index

The diagonal entries of $J^* P_{\theta^{\mathrm{MAP}}} J^{*\top}$ are also of interest and they are related to the variance in the noise-free output predictions. In particular, a relatively large ratio between the k-th diagonal entry of the matrix and $\hat{y}_k^*$ indicates an unreliable prediction at time instant k whose uncertainty is large compared to the predicted value itself. For a full sequence $\mathbf{u}^*$ of length N, we introduce in this paper an aggregate *surprise* index $s(\mathbf{u}^*)$ defined by:

$$s(\mathbf{u}^*) = 100 \times \frac{\sum_{k=0}^{N-1} \sqrt{\left(J^* P_{\theta^{\mathrm{MAP}}} J^{*\top}\right)_{kk}}}{\sum_{k=0}^{N-1} |\hat{y}_k^*(\theta^{\mathrm{MAP}})|} \ (\%), \tag{17}$$

which measures the relative size of the uncertainty throughout the sequence $\mathbf{u}^*$. In (17), the subscript kk denotes the diagonal element of a matrix in the k-th row and column.

It is important to note that computation of $s(\mathbf{u}^*)$ does not require the actual output $\mathbf{y}^*$ and thus it can be carried out even without running an actual experiment on the real system. Therefore, $s(\mathbf{u}^*)$ may be used to assess beforehand whether the model is expected to give reliable predictions when fed with the sequence $\mathbf{u}^*$.

5 Numerical example

In this section, we test the methodologies presented in the paper on a non-linear system identification problem. The developed software is based on the PyTorch deep-learning library and it is available in the GitHub repository `https://github.com/forgi86/sysid-neural-unc`. Computations are performed on a PC equipped with an AMD Ryzen 5 1600x processor, 32 GB of RAM, and an nvidia 1060 GPU.

We consider as true system a discrete-time Wiener-Hammerstein with sampling frequency $f_s = 51200$ Hz consisting in the series interconnection of a transfer function $G_1(z)$, a static non-linearity $f(\cdot)$, and a transfer function $G_2(z)$:

$$\begin{aligned}
G_1(z) &= \frac{0.010252 + 0.030757z^{-1} + 0.030757z^{-2} + 0.010252z^{-3}}{1 - 2.151941z^{-1} + 1.744729z^{-2} - 0.510767z^{-3}}\\
G_2(z) &= \frac{0.008706 - 0.004596z^{-1} - 0.004596z^{-2} + 0.008706z^{-3}}{1 - 2.574867z^{-1} + 2.235716z^{-2} - 0.652629z^{-3}}\\
f(x) &= \mathrm{elu}\left(-\frac{10}{11}x\right),
\end{aligned}$$

where $\mathrm{elu}(x) = e^{x-1}$ for $x \leq 0$ and 0 otherwise. The Bode plots of G_1, G_2 and the static non-linearity $f(\cdot)$ are also shown in Figure 1 and Figure 2, respectively. This Wiener-Hammerstein system is closely inspired to the dynamics of the benchmark Schoukens et al. (2009) involving a real electronic circuit. In this paper, we prefer this synthetic Wiener-Hammerstein system to the original benchmark in order to be able to generate data from different dynamical regimes with ease. For analogy with Schoukens et al. (2009), inputs and outputs of our numerical example are assumed to be in Volts (V) units hereafter.

As for the neural state-space model (1), in line with previously published results on the benchmark (Beintema et al., 2021), $\mathcal{F}$ and $\mathcal{G}$ have a single hidden layer with 15 nodes and tanh static non-linearity, plus a direct linear input/output term. In total, the model has $n_\theta = 385$ parameters.

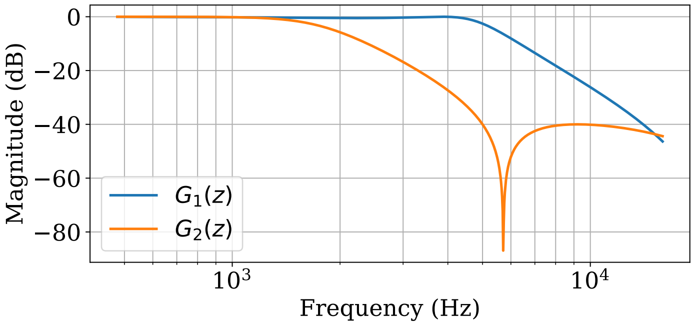

Figure 1: WH system: Bode diagrams of $G_1(z)$ and $G_2(z)$.

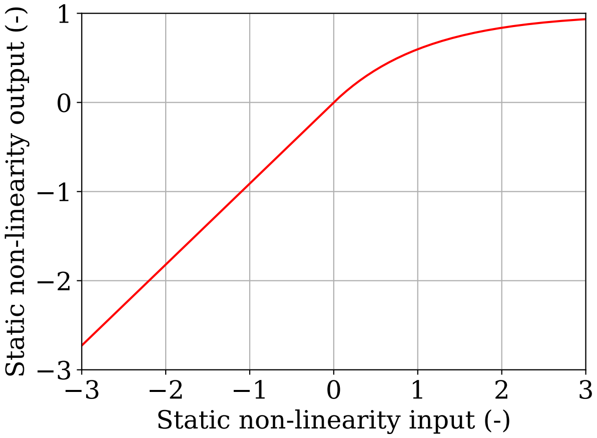

Figure 2: WH system: Static non-linearity $f(\cdot)$.

We use a training dataset where the input is a 10000-sample *multisine* signal with flat spectrum in the frequency range [0 2] kHz and standard deviation 0.4 V, and the output is corrupted by a white Gaussian noise with standard deviation $\sigma_e = {}^1\!/\!_\beta = 5 \cdot 10^{-3}$ V. Note that the input spectrum does not cover the *transmission zero* of the transfer function $G_2(z)$ located at approximately 5.5 kHz.

To compute the MAP estimate θ^{MAP} efficiently, the negative log-likelihood (6) is minimized over batches of sub-sequences extracted from the training data in random order, see Forgione and Piga (2020) for details. The batch size and sub-sequence length are both set to 256. Neural network parameter optimization is performed over 120 epochs[1] of the Adam algorithm followed by 4 epochs of L-BFGS, using the standard implementation and default settings of PyTorch. Overall, the optimization procedure takes 873 s.

Once θ^{MAP} is available, the posterior covariance $P_{\theta^{\mathrm{MAP}}}$ is obtained according to the Laplace approximation (12). The time required to obtain $P_{\theta^{\mathrm{MAP}}}$, which is largely dominated by the computation of the gradients $\frac{\partial \hat{y}_k}{\partial \theta}$, is 44 s using the recursive gradient computation method in Forgione et al. (2022), while it increases to 465 s with the naive implementation.

For model testing, we consider four scenarios where the input signal $\mathbf{u}^*$ is: 1) a multisine with standard deviation 0.4 V and bandwidth [0 2] kHz (same as training input); 2) a multisine with standard deviation 0.4 V and bandwidth [1 2] kHz; 3) a multisine with standard deviation 0.8 V and bandwidth [0 2] kHz; 4) a multisine with standard deviation 0.4 V and bandwidth [0 10] kHz. For each test set, we compute the nominal prediction $\hat{\mathbf{y}}^*$ by simulating the state-space model (1) with $\theta = \theta^{\mathrm{MAP}}$ and $e = 0$, and the approximate ppd according to (16). The approximate ppd is then used to obtain 99.7% credible intervals (having width ±3 times the square root of the diagonal entries of the approximate ppd's covariance matrix $\Sigma_{\mathbf{y}^*}$) and the surprise index $s(\mathbf{u}^*)$ according to (17).

We evaluate the performance of the nominal predictions in terms of the FIT index:

$$\mathrm{FIT} = 100 \times \left(1 - \frac{\sqrt{\sum_{k=0}^{N-1} \left(y_k^* - \hat{y}_k^*\right)^2}}{\sqrt{\sum_{k=0}^{N-1} \left(y_k^* - \overline{y}^*\right)^2}}\right) (\%), \tag{18}$$

where $\overline{y}^*$ is the sample mean of the sequence $\mathbf{y}^*$.

To evaluate the goodness of the credible intervals, we report their empirical *coverage*, namely the percentage of time steps where the actual output $\mathbf{y}^*$ lies inside the intervals. A value close to 99.7% indicates *well-calibrated* intervals.

Note that Signals 3 and 4 drive the system in a dynamical ranges unseen during training and thus force the model to operate in an extrapolation regime. For these signals, we expect the FIT index to decrease and the uncertainty intervals to get wider. Wider uncertainty bounds result in a larger surprise index $s(\mathbf{u}^*)$, which in turn should allow us to detect the FIT decrease without knowledge of the actual output $\mathbf{y}^*$.

[1] An epoch corresponds to the processing of all the contiguous sub-sequences of length 256 in the training dataset in random order.

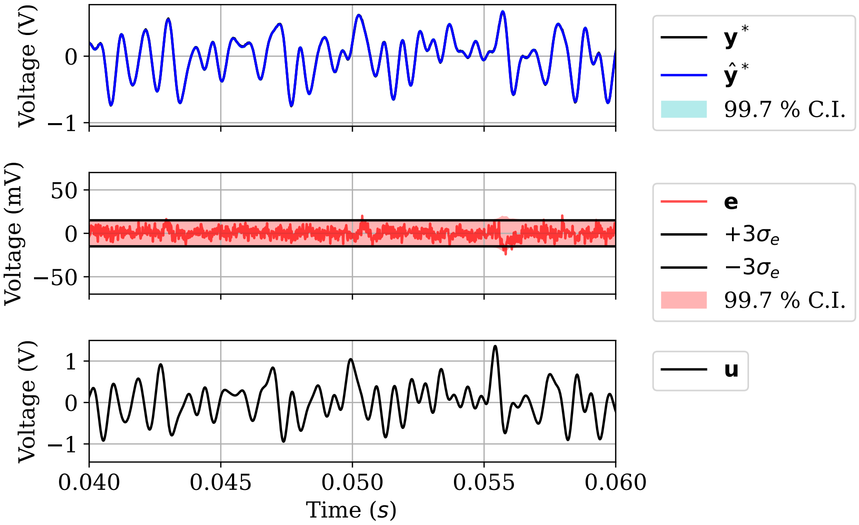

Figure 3: WH system: results on multisine signal 1.

The FIT index, surprise index, and coverage of the four test signals are reported in Table 1. We observe that FIT and surprise indexes are indeed negatively correlated, as expected.

In Figures 3, 4, 5, 6, we show relevant time traces for the four test signals. In the top panel, we show the actual output $\mathbf{y}^*$ (black line) together with the posterior mean $\hat{\mathbf{y}}^*$ (blue line) and 99.7% credible intervals (shaded blue area). In the middle panel, we show the error signal $\mathbf{e} = \mathbf{y}^* - \hat{\mathbf{y}}^*$ (red line), together with 99.7% credible intervals (shaded red area). We also show $\pm 3\sigma_e$ horizontal bands, corresponding to the aleatoric component of the output uncertainty (black lines). Finally, the bottom panel represents the input signal.

For Signals 1 and 2, the prediction quality is very high (black and blue line overlapping in the the top panel). Uncertainty bounds are not visible (as too narrow) in the top panel and can only be appreciated in the (magnified) middle one. In the middle panel, we also note that uncertainty bounds are largely dominated by the aleatoric component (red shaded area mostly comprised within the $\pm 3\sigma_e$ bands). Furthermore, these bounds are well calibrated (the red line lies within the bounds for most of the time steps), as also indicated by the coverage indexes which are close to the target value 99.7 %.

For Signal 3, the prediction quality decreases significantly in the time instant when the input/output samples are large (a condition not seen during training). The uncertainty bounds also expand in these regions and, remarkably, appear to be still rather well calibrated (i.e. the error $\mathbf{e}$ lies indeed within the uncertainty region in most of the time steps, with a coverage index of 96.1 %). Furthermore, the surprise index of 2.10 % is significantly larger than the one computed over

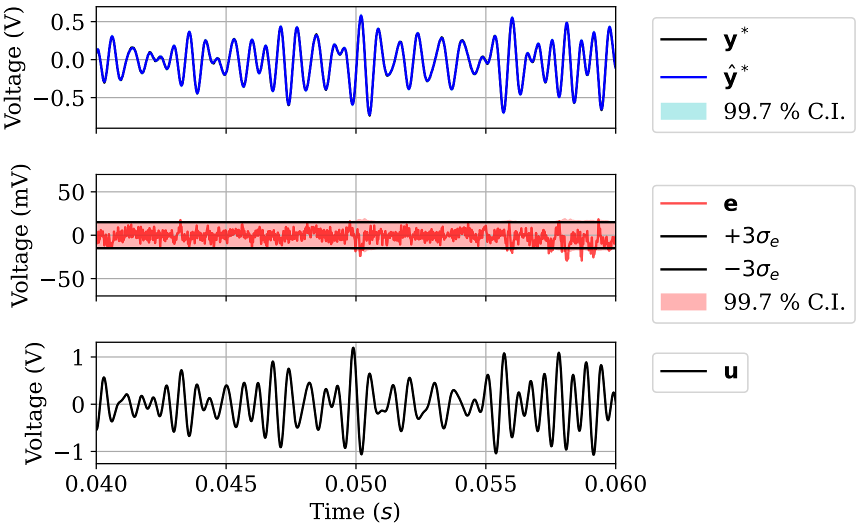

Figure 4: WH system: results on multisine signal 2.

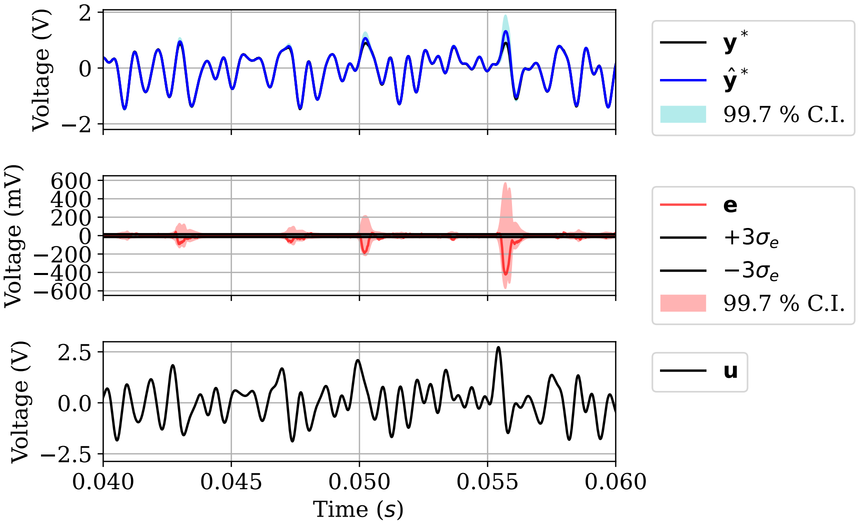

Figure 5: WH system: results on multisine signal 3.

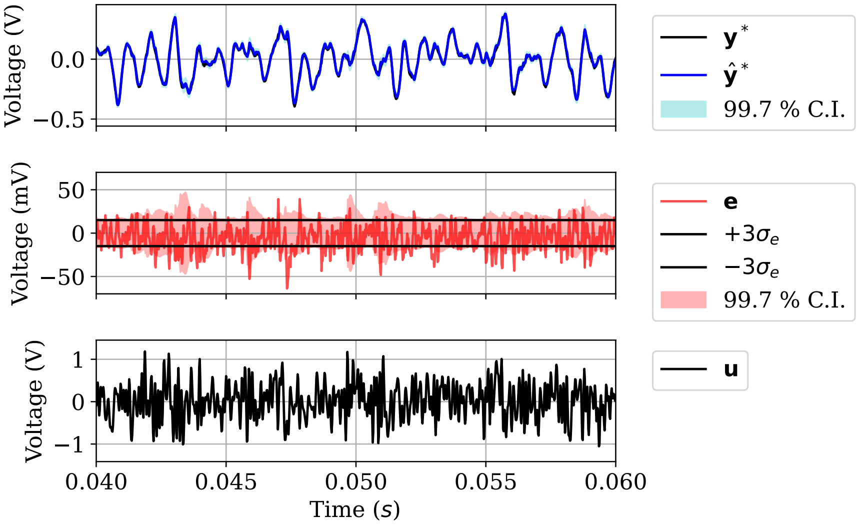

Figure 6: WH system: results on multisine signal 4.

Signal	FIT (%)	coverage (%)	surprise (%)
multisine 1	98.1	99.2	0.33
multisine 2	97.7	98.6	0.43
multisine 3	93.9	96.1	2.10
multisine 4	87.8	80.6	4.03

Table 1: FIT index, uncertainty intervals coverage, and surprise index on the test datasets.

Signals 1 and 2 (0.33 % and 0.43 %, respectively). Thus, the out-of-distribution regime is effectively detectable using the proposed methodology.

For Signal 4, uncertainty bounds are also enlarged, even though they are clearly not well calibrated. Indeed, the error signal $\mathbf{e}$ is too often out of the red shaded area, thus the latter is not a well-calibrated 99.7% credible interval, as also indicated by the low coverage of 80.6 %). Nonetheless, the high surprise index $s(\mathbf{u}^*) = 4.03$ alerts the used that the model is working in an extrapolation regime and consequently its performance will be low.

6 Conclusion

We have presented a viable approach for uncertainty quantification with neural state-space models. Based on the obtained uncertainty description, we have defined a surprise index that indicates whether the model predictions generated from a given input are expected to be reliable, i.e. close to the response of the true system.

This preliminary work may be extended in different directions. First, other inference approximation techniques may be adopted to obtain a richer and more accurate characterization of the uncertainty. In this sense, efficient sampling techniques such as Hamiltonian Monte Carlo may be considered to overcome the limiting assumptions (e.g. uni-modality) of the currently used Laplace approximation. A challenge in this sense is to devise scalable algorithms applicable to large neural-network models.

Furthermore, tools like the surprise index may be used to choose informative input signals to be used for model training/refinement. This could pave the way for experiment design and active learning in the context of system identification with neural state-space models.

Finally, to foster further research in uncertainty quantification and out-of-distribution recognition, specific benchmarks and performance metrics should be devised and shared with the system identification community.

Acknowledgement

This work was partially supported by the European H2020-CS2 project ADMITTED, Grant agreement no. GA832003.

References

Andersson, C., Ribeiro, A.H., Tiels, K., Wahlström, N., and Schön, T.B. (2019). Deep Convolutional Networks in System Identification. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 3670–3676. doi: 10.1109/CDC40024.2019.9030219.

Beintema, G., Tóth, R., and Schoukens, M. (2021). Nonlinear state-space identification using deep encoder networks. In *Learning for Dynamics and Control*, 241–250. PMLR.

Bishop, C.M. and Nasrabadi, N.M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

Forgione, M., Muni, A., Piga, D., and Gallieri, M. (2022). On the adaptation of recurrent neural networks for system identification. *arXiv preprint arXiv:2201.08660*.

Forgione, M. and Piga, D. (2020). Model structures and fitting criteria for system identification with neural networks. In *2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT)*, 1–6. doi:10.1109/AICT50176.2020.9368834.

Gawlikowski, J., Tassi, C.R.N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., et al. (2021). A Survey of Uncertainty in Deep Neural Networks. *arXiv preprint arXiv:2107.03342*.

Izmailov, P., Vikram, S., Hoffman, M.D., and Wilson, A.G.G. (2021). What are Bayesian Neural Network Posteriors Really Like? In *International conference on machine learning*, 4629–4640. PMLR.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *Advances in Neural Information Processing Systems*, 30.

Ljung, L., Andersson, C., Tiels, K., and Schön, T.B. (2020). Deep Learning and System Identification. *IFAC-PapersOnLine*, 53(2), 1175–1181.

Loquercio, A., Segu, M., and Scaramuzza, D. (2020). A General Framework for Uncertainty Estimation in Deep Learning. *IEEE Robotics and Automation Letters*, 5(2), 3153–3160.

Maddox, W.J., Izmailov, P., Garipov, T., Vetrov, D.P., and Wilson, A.G. (2019). A Simple Baseline for Bayesian Uncertainty in Deep Learning. *Advances in Neural Information Processing Systems*, 32.

Mavkov, B., Forgione, M., and Piga, D. (2020). Integrated neural networks for nonlinear continuous-time system identification. *IEEE Control Systems Letters*, 4(4), 851–856.

Peeters, L., Beintema, G.I., Forgione, M., and Schoukens, M. (2022). NARX identification using Derivative-Based Regularized Neural Networks. *arXiv preprint arXiv:2204.05892*.

Schoukens, J., Suykens, J., and Ljung, L. (2009). Wiener-Hammerstein benchmark. In *Proc. of the 15th IFAC symposium on System Identification (SYSID 2009)*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1), 1929–1958.

Van den Bos, A. (2007). *Parameter estimation for scientists and engineers.* John Wiley & Sons.

Wilson, A.G. and Izmailov, P. (2020). Bayesian Deep Learning and a Probabilistic Perspective of Generalization. *Advances in neural information processing systems*, 33, 4697–4708.

Wright, S., Nocedal, J., et al. (1999). Numerical optimization. *Springer Science*, 35(67-68), 7.

Wu, R.T. and Jahanshahi, M.R. (2019). Deep Convolutional Neural Network for Structural Dynamic Response Estimation and System Identification. *Journal of Engineering Mechanics*, 145(1), 04018125.

Zhou, H., Ibrahim, C., Zheng, W.X., and Pan, W. (2022). Sparse Bayesian Deep Learning for Dynamic System Identification. *Automatica*, 144, 110489.