Nonlinear state-space identification using deep encoder networks

Proceedings of Machine Learning Research vol 144:1–10, 2021

Nonlinear state-space identification using deep encoder networks

Gerben Beintema G.I.BEINTEMA@TUE.NL
Roland Toth R.TOTH@TUE.NL
Maarten Schoukens M.SCHOUKENS@TUE.NL
Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB, The Netherlands

Abstract

Nonlinear state-space identification for dynamical systems is most often performed by minimizing the simulation error to reduce the effect of model errors. This optimization problem becomes computationally expensive for large datasets. Moreover, the problem is also strongly non-convex, often leading to sub-optimal parameter estimates. This paper introduces a method that approximates the simulation loss by splitting the data set into multiple independent sections similar to the multiple shooting method. This splitting operation allows for the use of stochastic gradient optimization methods which scale well with data set size and has a smoothing effect on the non-convex cost function. The main contribution of this paper is the introduction of an encoder function to estimate the initial state at the start of each section. The encoder function estimates the initial states using a feed-forward neural network starting from historical input and output samples. The efficiency and performance of the proposed state-space encoder method is illustrated on two well-known benchmarks where, for instance, the method achieves the lowest known simulation error on the Wiener–Hammerstein benchmark.

Keywords: Nonlinear System Identification, Deep Learning, State-Space, Multiple Shooting.

1. Introduction

Linear system identification is already well developed, in both a theoretical and a practical sense. However, due to increasing performance demands, the use of light-weight materials and/or increased demand for energy efficiency, linear identification and control falls short of meeting these demands and hence, nonlinear system identification and control has become increasingly important. This paper introduces a novel nonlinear state-space identification approach that reduces the computational cost by combining techniques and insights of machine learning and dynamical system identification.

Computational tractability is often hard to achieve while estimating nonlinear (state-space) models when minimizing the simulation error. Nevertheless, the use of simulation error is essential in practise to increase model reliability when model errors are present (Schoukens and Ljung, 2019). However, the simulation error objective is commonly computationally intractable often caused by the lack of smoothness of the loss function and/or the gradients of the loss function (Ribeiro et al., 2020). Furthermore, the computational cost of calculating the loss scales linearly with the length of the measured time-series. This limits the applications to relatively small datasets and is a serious detriment in the age of big data. Moreover, it has been shown that artificial neural network are powerful function approximators when large datasets are applied (Chiroma et al., 2018).

Multiple methods exist which aim to negate these causes of intractability: (*i*) Careful initialization of the model parameters (Schoukens and Toth, 2020), resulting in an initial estimate which is expected to be close to the global minimum of the loss function and avoids gradient and system

instability, and (*ii*) the multiple shooting method (Bock, 1981) which splits the time series into multiple sections where each section has its own independent loss function. This splitting operation has recently been shown to have a smoothing effect on the loss function and its gradient (Ribeiro et al., 2020) making gradient based optimization method easier to apply.

How to estimate the initial state at the start of each section remains one of the main issues in successfully applying the multiple shooting method. Two approaches are commonly used: (*i*) Setup the initial states as parameters of the optimization (Bock, 1981), this however scales the model complexity with the number of sections, and (*ii*) estimate the initial state by using equality constrains to the final state of the previous section (Ribeiro et al., 2020), this constraint optimization is considerably more involved.

This paper proposes a new approach to the initialization problem by using an encoder function. This function estimates the current state based on historical inputs and outputs. The use of an encoder function in combination with a multi-step ahead prediction loss can be viewed as an extension of sub-space identification approaches such as CCA (Katayama, 2006). The proposed approach reduces the transient errors and does not increase the model complexity with the number of sections and, moreover, it provides generalization to other unseen datasets as the encoder allows one to jump start the simulation at the correct model state. Furthermore, in this paper we demonstrate that the state-space encoder method with artificial neural network achieves state-of-the-art performance on the Wiener–Hammerstein benchmark (Schoukens et al., 2009) and obtains a competitive performance on the Silverbox benchmark (Wigren and Schoukens, 2013) significantly exceeding previously proposed deep learning methods on these benchmarks.[1]

The remainder of this paper first discusses and motivates the state-space encoder method in Section 2. Next, the method is applied to two well-known benchmarks and the results are compared quantitatively with other results from literature in Section 3 followed by a discussion of the presented approach in Section 4.

2. Encoder-Based nonlinear state-space identification

2.1. State-space model structure

The following discrete-time model structure is considered:

$$\hat{x}_{t+1} = f_\theta(\hat{x}_t, u_t), \tag{1a}$$

$$\hat{y}_t = h_\theta(\hat{x}_t, u_t), \tag{1b}$$

where $t \in \mathbb{Z}$ is the discrete time index, $x \in \mathbb{R}^{n_x}$ the internal state vector, $u_t \in \mathbb{R}^{n_u}$ and $y_t \in \mathbb{R}^{n_y}$ are the model input and output respectively, θ the model parameters, and f_θ, h_θ are the nonlinear dynamics of the state-space model. We assume that the measured data is generated by a system contained within this model class: $y_t = h_{\theta_0}(x_t, u_t) + v_t$ and $x_{t+1} = f_{\theta_0}(x_t, u_t)$, where $v_t \in \mathbb{R}^{n_y}$ is zero-mean (possibly coloured) noise with finite variance.

2.2. Classical simulation error identification

A common approach to estimate the model parameters is to minimize the simulation error as

$$V_{\text{simulation}}(\theta) = \frac{1}{N_{\text{samples}}} \sum_{t=1}^{N_{\text{samples}}} ||h_\theta(x_t, u_t) - y_t||_2^2. \tag{2}$$

1. Code available at https://github.com/GerbenBeintema/SS-encoder-WH-Silver

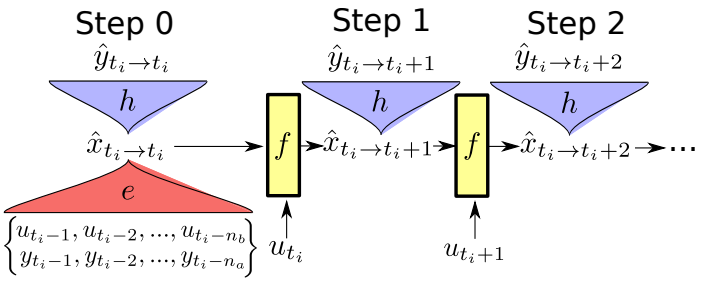


Figure 1: The state-space encoder method applied on a section of a time series starting at t_i where the initial state $\hat{x}_{t_i \to t_i}$ is estimated using the encoder function e using the historical inputs and outputs.

where x_t dependent on f_θ. For convenience of writing use a single parameter vector θ for both functions, however, in practise they are independently parameterized. However, using this expression directly is often challenging. Each evaluation of $V_{\text{simulation}}(\theta)$ requires $O(N_{\text{samples}})$ operations in series which makes it intractable for large data sets. Moreover, it is known empirically and shown theoretically that this expression can result in many local minima and in unstable behaviour of the optimization for gradient based techniques (Ribeiro et al., 2020).

2.3. Encoder networks for nonlinear identification

To negate the problems observed for the simulation loss we employ a loss function that sums over N independent sections of the data with starting index t_i and length $T + k_0 + 1$ similar to the multiple shooting method which is known to have a stabilizing and smoothing effect (Ribeiro et al., 2020). The full formulation of the proposed state-space encoder method is given by:

$$V_{\text{encoder}}(\theta) = \frac{1}{2N(T+1)} \sum_{i=1}^{N} \sum_{k=k_0}^{T+k_0} ||\hat{y}_{t_i \to t_i+k} - y_{t_i+k}||_2^2, \tag{3a}$$

$$\hat{y}_{t_i \to t_i+k} := h_\theta(\hat{x}_{t_i \to t_i+k}, u_{t_i+k}), \tag{3b}$$

$$\hat{x}_{t_i \to t_i+k+1} := f_\theta(\hat{x}_{t_i \to t_i+k}, u_{t_i+k}), \tag{3c}$$

$$\hat{x}_{t_i \to t_i} := e_\theta(y_{t_i-n_a:t_i-1}, u_{t_i-n_b:t_i-1}), \tag{3d}$$

where $t_i \to t_i + k$ reads as "The simulated state at $t_i + k$ starting at t_i with initial state $\hat{x}_{t_i \to t_i}$". Furthermore, the choice of $k_0 \geq 0$ allows for an initial transient time to be excluded from the loss calculation. Finally, to close this expression, an encoder function e_θ is introduced to estimate the initial state starting from historical input and output samples $u_{t-n_b:t-1} \in \mathbb{R}^{n_u \cdot n_b}$ and $y_{t-n_a:t-1} \in \mathbb{R}^{n_u \cdot n_a}$. A graphic representation of the state-space encoder method is shown in Figure 1.

This approach and multiple shooting is related to Truncated Backpropagation Through Time (TBTT) (Tallec and Ollivier, 2017) which is a gradient computation method which truncates the gradient calculations after a few backwards steps. This, however, requires a initial pass over the entire dataset length which similarly to simulation error loss (Equation (2)) scales the computational complexity with dataset length. Moreover, Our approach and multiple shooting works on the level of the cost function whereas TBTT is a gradient computation method.

This expression can be interpreted as a trade-off between simulation error with $N = 1$, $T = N_{samples}$ and prediction error with $T = 0$, $N = N_{samples}$, $k_0 = 0$ under the right assumptions. Note that sections can overlap. This is normally excluded in multiple shooting approaches, but is allowed in the approach presented in this paper. Neural network structures (e.g. fully connected neural networks, convolutional neural networks) are used to represent the encoder function due to their excellent function approximation abilities for large data sets (Poggio et al., 2017).

This approach has a few computational advantages over the two existing state initialization methods when considering the multiple shooting method context. As mentioned in the introduction, a first method is to initialize as $\hat{x}_{t_i \to t_i} = 0$ (Bock, 1981) which often requires a system-specific burn time k_0 such that the transient is sufficiently suppressed. This burn time can significantly increase the computational cost when a long or even infinite transient (e.g. resonating and chaotic systems) is present. The second method includes $\hat{x}_{t_i \to t_i}$ as a model parameter (Ribeiro et al., 2020). However, this significantly increases the model complexity by adding $N \cdot n_x$ parameters. Moreover, this method also provides no generalization to other datasets. The proposed encoder method negates the above mentioned computational hindrances and potentially generalizes to new data sets.

The state-space encoder method also connects with the sub-space identification literature. Sub-space identification uses an encoder and a decoder map to identify a sub-space for a dynamical system such as CCA (Katayama, 2006). This approach can be extended to the proposed method by choosing the structure of the decoder map to an unrolled state-space model as in Figure 1. This connection will be further explored in future research. Moreover, the proposed method is also arguably simpler than the closely related auto-encoder approach (Masti and Bemporad, 2018) for it only employs a single loss function and it allows for the minimization of a multi-step criterion.

2.4. Batch optimization

Adapting the cost framework proposed in Equation (3d) allows the loss to be calculated independently on each section. Firstly, this independence allows for close to trivial parallelization, resulting in a reduced computational cost on modern hardware. Secondly, one can choose to sum not over all sections but only a subset of possible sections. This results in a batch loss formulation of the multiple shooting method as:

$$V_{\text{batch}}(\theta) = \frac{1}{2N_{\text{batch}}(T+1)} \sum_{i \in B} \sum_{k=k_0}^{T+k_0} ||\hat{y}_{t_i \to t_i+k} - y_{t_i+k}||^2, \tag{4a}$$

$$B \subset \{1, 2, ..., N\}. \tag{4b}$$

The batch loss formulation allows one to utilize modern powerful batch optimization algorithms developed by the machine learning community (e.g. Adam (Kingma and Ba, 2014)) that scales well for increasing data set size.

3. Numerical experiments

In this section, the real-world modeling performance of the state-space encoder method is analyzed by applying the proposed method to two well-known system identification benchmarks: The Wiener–Hammerstein and the Silverbox benchmark.[2] The obtained results are compared quantitatively with other results listed in the literature.

2. Data obtained from https://sites.google.com/view/nonlinear-benchmark/

3.1. The Wiener–Hammerstein benchmark

The Wiener–Hammerstein benchmark (Schoukens et al., 2009) is implemented as an electronic circuit with a diode-resistor nonlinearity (SISO). This benchmark consists of 80,000 training samples, 20,000 validation samples and 78000 test samples.

The encoder function e_θ and both dynamics function f_θ and h_θ are all represented using a single hidden layer neural network with 15 hidden nodes and tanh activation functions. Furthermore, this structure also includes a parallel linear function that goes directly from the input of the neural network to the output of the network without any nonlinear activation functions similar to a residual layer (He et al., 2016):

$$\mathbf{z}_{\text{out}} = \mathbf{A}_1 \tanh(\mathbf{A}_2 \mathbf{z}_{\text{in}} + \mathbf{b}_2) + \mathbf{A}_3 \mathbf{z}_{\text{in}} + \mathbf{b}_3 \tag{5}$$

with $\mathbf{z}_{\text{in}}$ being the network input in vector form, $\mathbf{z}_{\text{out}}$ the network output and $\mathbf{A}_i$ and $\mathbf{b}_i$ the parameters of the network. These parameters are initialized by sampling from the uniform distribution $\mathcal{U}(-\sqrt{k}, \sqrt{k})$ with $k = 1/\sqrt{n_{\text{in}}}$ where n_{in} are the number of inputs (i.e. number of elements in $\mathbf{z}_{\text{in}}$)

The Wiener–Hammerstein benchmark encoder state-space model structure has the following settings $n_x = 6$ (equal to the underlying system order), $k_0 = 0$ (no transient corrections required), $T = 80$ (taken approximately four times the time scale of the system which is approximately 20 steps), $n_a = n_b = 50$ (larger than n_x and a few time the time scale). Furthermore, a 32-bit floating-point accuracy is used for the parameter estimation. The multiple shooting starting points t_i can be any possible starting point within the range of the training set to maximally use the available data. This allows for overlapping training sections. Furthermore, the Adam batch optimization method (Kingma and Ba, 2014) is utilized with a learning rate of $\alpha = 10^{-3}$ and a batch size of 1024 which adjusts the learning rate based on the variance of the gradient. During optimization the performance of the estimated model is evaluated by monitoring the simulation error on the validation set after each epoch. The estimated model is saved if a new lowest simulation error on the validation set has been achieved (i.e. early stopping). Furthermore, both the input and the output are normalized by subtracting the mean and dividing by standard deviation to improve the performance and training time. Lastly, after the batch training converged, a local minimum search using all training data is performed. However, this has only improved the final result marginally (i.e. 0.101% test NRMS without final search).

The model performance is reported in both Root Mean Square (RMS) and the Normalized Root Mean Square (NRMS) of the simulation error:

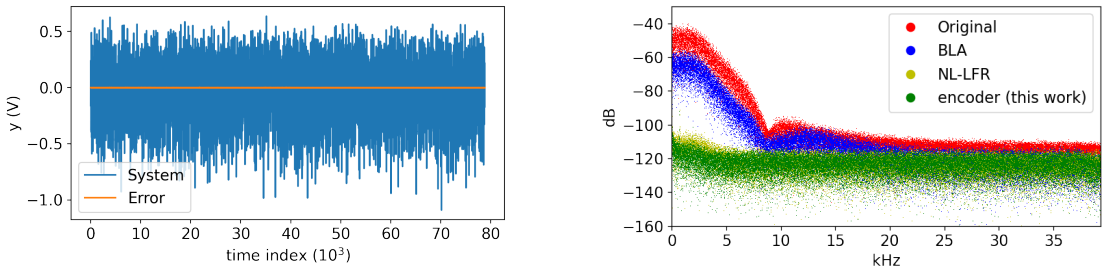
$$\text{NRMS} = \frac{\sqrt{1/N \sum_{t=t_0}^{N+t_0} ||\hat{y}_t - y_t||_2^2}}{\sigma_y} = \frac{\text{RMS}}{\sigma_y} \tag{6}$$

with $\sigma_y = 244.7\ \text{mV}$ the standard deviation of the measured test output.

The results obtained on the Wiener–Hammerstein benchmark are reported in Table 1. The table shows that the proposed encoder method has, to the author's knowledge, the best known RMS simulation error reported in the literature for this benchmark. Furthermore, one can see in Figure 2 that the remaining error in the time domain is visually a straight line and the remaining error in the frequency domain is reduced significantly compared to error obtained by the best linear approximation. Also note that other models with larger neural networks and higher state dimension n_x resulted in a similar model performance (e.g. with $n_x = 8$ and two hidden layer neural networks yielded a test NRMS simulation of 0.1011%). This insensitivity to the model structure setting indicates that

Table 1: Performance of the state-space encoder on the Wiener–Hammerstein benchmark compared to the results reported in the literature.

Identification Method	Test RMS Simulation (mV)	Test NRMS Simulation
State-space Encoder (this work)	**0.241**	**0.0987%**
QBLA (Schoukens et al., 2014)	0.279	0.113%
Pole-zero splitting (Sjöberg et al., 2012)	0.30	0.123%
NL-LFR (Schoukens and Toth, 2020)	0.30	0.123%
PNLSS (Paduart et al., 2012)	0.42	0.172%
Generalized WH (Wills and Ninness, 2009)	0.49	0.200%
LS-SVM (Falck et al., 2009)	4.07	1.663%
Bio-social evolution (Naitali and Giri, 2016)	8.55	3.494%
Auto-encoder (reproduction) (Masti and Bemporad, 2018)	12.01	4.907%
Genetic Programming (Khandelwal, 2020)	23.50	9.605%
SVM (Marconato and Schoukens, 2009)	47.40	19.373%
BLA (Lauwers et al., 2009)	56.20	22.969%



(*a*) Time domain (*b*) Frequency domain

Figure 2: The simulation error of the state-space encoder method evaluated on the test set of the Wiener–Hammerstein benchmark in both time and frequency domain.

the importance of careful model structure selection is reduced for the state-space encoder method when being used in combination with large data sets. Note that the generalization gap (i.e. the gap between training error and test error) is negligible. The NRMS simulation error on the training set is 0.09789% and on the test set is 0.09870%. This indicates that virtually no overfitting is taking place. Also observe that these results are obtained using random initial parameters while many of the approaches listed in Table 1 require a linear model estimate or other parameter initialization schemes to obtain competitive results.

The NRMS error during optimization is shown in Figure 3. It can be observed that the introduction of encoder-based batch optimalization does not only improve the training time, but also significantly improves the model quality. Though, even with the improvements in training speed introduced by utilizing multiple shooting, the encoder and batch optimization, the training still takes considerable time. The optimization took $4 \cdot 10^4$ epochs and $4 \cdot 10^6$ batch updates. However, a high-quality model is already obtained, before the optimization method fully converges, with only a tenth of the time budget. This can possibly be further improved in the future by using more ap-

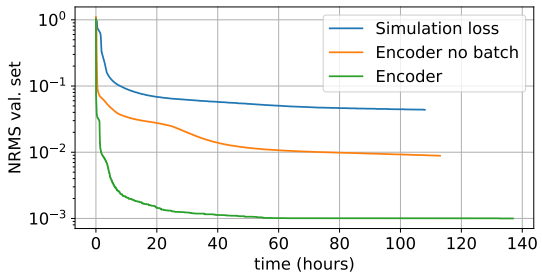


Figure 3: The lowest NRMS simulation error on the validation set during training for the Wiener–Hammerstein benchmark with references of the simulation error approach (Equation 2) and using the encoder method without batch optimization.

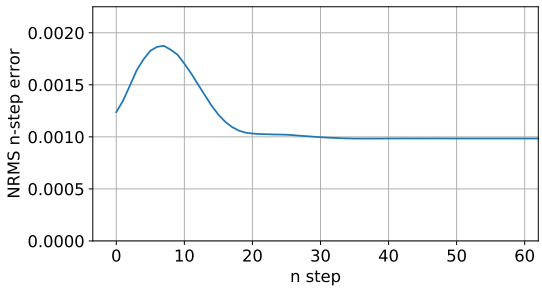


Figure 4: The n-step NRMS as in Equation (7) on the test set of the Wiener–Hammerstein benchmark.

propriate optimization algorithms such as the recently proposed quasi-Newton methods (Wills and Schön, 2019).

The n-step NRMS error is introduced to get some insight into how well the encoder can estimate the initial state and to validate the choice of hyper-parameters $k_0 = 0$ and $T = 80$. The n-step NRMS error is introduced as the normalized error you expect so find after taking n steps as in:

$$\text{NRMS}_n = \frac{\sqrt{1/M \sum_{i=1}^{M} (\hat{y}_{t_i \to t_i+n} - y_{t_i+n})^2}}{\sigma_y}. \tag{7}$$

This quantity is shown in Figure 4. The average is taken over all the possible starting t_i of the test set. In the figure, one can see that the encoder does not provide a perfect estimate of the initial state as indicated by a bump that is seen for $n < 30$ with the peak being at $n = 7$. Nevertheless, this does validate the choice of $T = 80$ and $k_0 = 0$ as the transient does not dominate the loss function.

3.2. The Silverbox benchmark

The Silverbox system benchmark (Wigren and Schoukens, 2013) is an electronic implementation of a mass-spring-damper system with a nonlinear spring, i.e. a forced Duffing oscillator. The Silverbox system can be modeled by a second-order nonlinear state-space model. The data set consists of *test* section of 40000 samples consisting of a filtered Gaussian excitation with slowly increasing amplitude whereas the *train* section of 87000 samples consists of the same filtered Gaussian excitation with constant amplitude where the last 30000 samples are also used to monitor the performance. The remaining 21000 samples composes the validation set. Note that the constant amplitude of the train and validation sections is smaller than the highest amplitude present in the test section requiring good extrapolation properties during testing.

The state-space encoder setup for the Silverbox benchmark is similar to the setup for the Wiener–Hammerstein benchmark. A 2 hidden layer neural network with 64 nodes per layer, $\tanh$ activa-

Table 2: Performance of the state-space encoder method on the Silverbox benchmark compared with literature. The numbers in parentheses indicate test set evaluation excluding extrapolation.

Identification Method	Val. RMS simulation (mV)	Test RMS simulation (mV)
PNLSS (Paduart et al., 2010)	–	0.26
Sigmoidal network models (Ljung et al., 2004)	–	0.3
LS-SVM (Espinoza et al., 2004)	0.23	0.32
Poly-LFR (Van Mulders et al., 2013)	–	0.35
Genetic Programming (Khandelwal, 2020)	0.09	0.36
Direct Identification (Hjalmarsson and Schoukens, 2004)	1.4	0.96
Local linear models (Verdult, 2004)	1.1	1.3
State-space Encoder with $n_x = 4$ (this work)	**0.36**	**1.4 (0.32)**
State-space Encoder with $n_x = 2$ (this work)	**1.0**	**2.4 (0.83)**
Best Linear Approximation (Marconato et al., 2012)	6.9	13.5

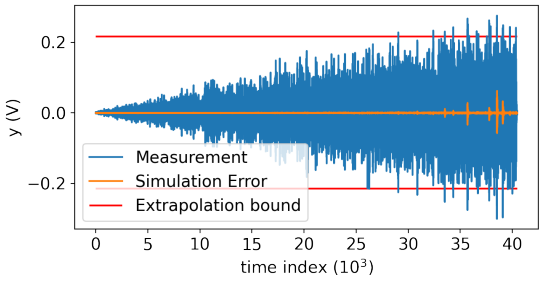


Figure 5: The remaining simulation error obtained using the encoder method ($n_x = 4$) on the test set of the Silverbox benchmark. When the output exceeds the training values (i.e. extrapolation) the simulation error increases significantly.

tions, and a linear bypass for e_θ, f_θ and h_θ is used. The encoder hyperparameters are set as: $k_0 = 0$, $T = 100$, $n_a = n_b = 50$, a batch size of 256, and using the Adam optimizer with the learning rate $\alpha = 10^{-3}$. The multiple shooting starting point can again be any possible starting point within the range of the training set.

A summary of the results is reported in Table 2. An observation is that taking the number of internal states equal to the number of real internal states $n_x = 2$ performs almost 3 times worse then taking $n_x = 4$. This could be due to the local minima being more pronounced at low state orders. The performance of the state-space encoder method is significantly worse on the test set when comparing to the other methods. However, this can be almost entirely be attributed to the extrapolation errors as can be observed in Figure 5. Furthermore, observe that almost all the state-of-the-art models use a polynomial representation of the nonlinearity, which matches with the true system structure. However, the method presented used a neural network to model the nonlinear function which introduced larger extrapolation errors. When the region of extrapolation is excluded from the test set the error drops in the range of state-of-the-art performance.

4. Discussion

This paper presented a novel nonlinear system identification approach realized by combining ideas and methods from machine learning, multiple shooting and subspace identification. The introduction of an encoder function that estimates the internal state based on the historical input and output data, together with multiple computational improvements, such as batch optimization, results in a computationally efficient nonlinear identification method that scales well to large datasets. Without requiring specific parameter initialization approaches (random parameter initialization has been used), the method was able to obtain the best known performance on the Wiener–Hammerstein benchmark. However, currently a known drawback of the encoder method is that the hyper-parameters are system dependent and require manual tuning. A detailed theoretical analysis and further computational improvements of the proposed identification method are the subject of future work.

References

H.G. Bock. Numerical treatment of inverse problems in chemical reaction kinetics. In *Modelling of chemical reaction systems*, pages 102–125. Springer, 1981.

Haruna Chiroma, Usman Ali Abdullahi, Ala Abdulsalam Alarood, Lubna A Gabralla, Nadim Rana, Liyana Shuib, Ibrahim Abaker Targio Hashem, Dada Emmanuel Gbenga, Adamu I Abubakar, Akram M Zeki, et al. Progress on artificial neural networks for big data analytics: A survey. *IEEE Access*, 7:70535–70551, 2018.

M. Espinoza, K. Pelckmans, L. Hoegaerts, J.A.K. Suykens, and B. De Moor. A comparative study of LS-SVM's applied to the silver box identification problem. *IFAC Proceedings Volumes*, 37 (13):369–374, 2004.

T. Falck, K. Pelckmans, J.A.K. Suykens, and B. De Moor. Identification of Wiener–Hammerstein systems using LS-SVMs. *IFAC Proceedings Volumes*, 42(10):820–825, 2009.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

H. Hjalmarsson and J. Schoukens. On direct identification of physical parameters in non-linear models. *IFAC Proceedings Volumes*, 37(13):375–380, 2004.

T. Katayama. *Subspace methods for system identification*. Springer Science & Business, 2006.

D. Khandelwal. *Automating data-driven modelling of dynamical systems: an evolutionary computation approach*. PhD thesis, Eindhoven University of Technology, 2020.

D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

L. Lauwers, R. Pintelon, and J. Schoukens. Modelling of Wiener–Hammerstein systems via the best linear approximation. *IFAC Proceedings Volumes*, 42(10):1098–1103, 2009.

L. Ljung, Q. Zhang, P. Lindskog, and A. Juditsky. Modeling a non-linear electric circuit with black box and grey box models. In *IFAC Symposium on Nonlinear Control Systems, Stuttgart, Germany, September*, 2004.

A. Marconato and J. Schoukens. Identification of Wiener–Hammerstein benchmark data by means of support vector machines. *IFAC Proceedings Volumes*, 42(10):816–819, 2009.

A. Marconato, J. Sjöberg, J. Suykens, and J. Schoukens. Identification of the silverbox benchmark using nonlinear state-space models. *IFAC Proceedings Volumes*, 45(16):632–637, 2012.

D. Masti and A. Bemporad. Learning nonlinear state-space models using deep autoencoders. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 3862–3867, 2018.

A. Naitali and F. Giri. Wiener–Hammerstein system identification–an evolutionary approach. *International Journal of Systems Science*, 47(1):45–61, 2016.

J. Paduart, L. Lauwers, J. Swevers, K. Smolders, J. Schoukens, and R. Pintelon. Identification of nonlinear systems using polynomial nonlinear state space models. *Automatica*, 46(4):647–656, 2010.

J. Paduart, L. Lauwers, R. Pintelon, and J. Schoukens. Identification of a Wiener–Hammerstein system using the polynomial nonlinear state space approach. *Control Engineering Practice*, 20 (11):1133–1139, 2012.

T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *International Journal of Automation and Computing*, 14(5):503–519, 2017.

A.H. Ribeiro, K. Tiels, J. Umenberger, T.B. Schön, and L.A. Aguirre. On the smoothness of nonlinear system identification. *Automatica*, 121:109158, 2020.

J. Schoukens and L. Ljung. Nonlinear system identification: A user-oriented road map. *IEEE Control Systems Magazine*, 39(6):28–99, 2019.

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

M. Schoukens and R. Toth. On the initialization of nonlinear LFR model identification with the best linear approximation. *IFAC 2020 World Congress, Berlin, Germany. 12 - 17 july*, 2020.

M. Schoukens, R. Pintelon, and Y. Rolain. Identification of Wiener–Hammerstein systems by a nonparametric separation of the best linear approximation. *Automatica*, 50(2):628–634, 2014.

J. Sjöberg, L. Lauwers, and J. Schoukens. Identification of Wiener–Hammerstein models: Two algorithms based on the best split of a linear model applied to the sysid'09 benchmark problem. *Control Engineering Practice*, 20(11):1119–1125, 2012.

Corentin Tallec and Yann Ollivier. Unbiasing truncated backpropagation through time. *arXiv preprint arXiv:1705.08209*, 2017.

A. Van Mulders, J. Schoukens, and L. Vanbeylen. Identification of systems with localised nonlinearity: From state-space to block-structured models. *Automatica*, 49(5):1392–1396, 2013.

V. Verdult. Identification of local linear state-space models: the silver-box case study. *IFAC Proceedings Volumes*, 37(13):393–398, 2004.

T. Wigren and J. Schoukens. Three free data sets for development and benchmarking in nonlinear system identification. In *2013 European control conference (ECC)*, pages 2933–2938, 2013.

A. Wills and B. Ninness. Estimation of generalised Hammerstein–Wiener systems. *IFAC Proceedings Volumes*, 42(10):1104–1109, 2009.

A. Wills and T.B. Schön. Stochastic quasi-Newton with line-search regularization. *arXiv preprint arXiv:1909.01238*, 2019.