Received: 5 June 2020 | Accepted: 24 November 2020

DOI: 10.1002/pamm.202000036

Nonlinear System Identification of a Furuta Pendulum Using Machine Learning Techniques

Tobias Rückwald[1,*], **Svenja Drücker**[1], **Daniel-André Dücker**[1], and **Robert Seifried**[1]

[1] Institute of Mechanics and Ocean Engineering, Hamburg University of Technology

Usually, dynamical systems can be described by differential equations. An accurate model is essential when designing and optimizing a controller. However, not every system can be modeled easily by physical models due to highly nonlinear behavior, such as friction or backlash. Then, a data based approach, such as machine learning, might be helpful. The focus in this work is set on modeling dynamical systems using neural networks and deep learning, which are growing subjects in research and industry to identify nonlinear dynamics.

1 Introduction

A Furuta pendulum, shown in Fig. 1, acts as a test plant in this work. It consists of two elements: a rotating arm, which can be actuated by the torque τ, and a non-actuated pendulum at the upper end of the arm. As displayed in Fig. 2, θ_1 describes the angular orientation of the arm and θ_2 the angular orientation of the pendulum. The state vector $\boldsymbol{x}$ is given by

$$\boldsymbol{x} = \begin{bmatrix} \theta_1 & \theta_2 & \dot{\theta}_1 & \dot{\theta}_2 \end{bmatrix}^\top . \tag{1}$$

The arm is driven by a motor via a transmission. This allows the use of a lower powered motor but introduces a highly nonlinear element to the system, due to friction and gear backlash of the transmission.

Fig. 1: Furuta pendulum.

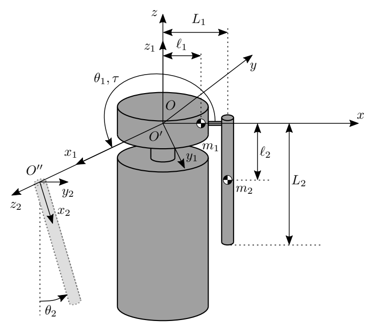

Fig. 2: Model of the Furuta pendulum.

Deep learning is a data based approach, hence a number of experiments are carried out. These time series should then be predicted by neural networks. In this work, three modeling approaches are presented and compared in terms of the prediction accuracy. As a reference, the equation of motion of the Furuta pendulum is used, whereby the transmission ratio of 14 to 1 and the static friction are included. Further nonlinear effects in the drive train are not modeled. The system constants are determined in a parameter study.

2 Methods for Nonlinear System Identification, Data Generation and Training

Firstly, the neural network autoregressive model with exogenous input (NNARX), which is a simple feedforward neural network, is used [3]. In a NNARX model, the state vectors $\boldsymbol{x}$ and system input vectors $\boldsymbol{u}$ of n previous time steps are fed into the neural network. On this basis the next time step is predicted.

Secondly, long short-term memory (LSTM) models are tested, which are well suited for sequence modeling, due to an internal memory [1]. This architecture is recurrent and more complex in its structure.

Thirdly, a hybrid approach consisting of the equation of motion and a neural network is demonstrated. The differential equation models the overall dynamics of the system and the unmodeled part is estimated by the neural network on acceleration level. The resulting dynamics are then integrated with the Runge–Kutta fourth-order method. Since the acceleration cannot be measured directly, it is determined by a post processing filter, the minimum model error (MME) estimator [2]. The input-output behavior of the filter is then learned by the neural network to achieve real-time capability.

* Corresponding author: e-mail tobias.rueckwald@tuhh.de, phone +49 40 42878 4012

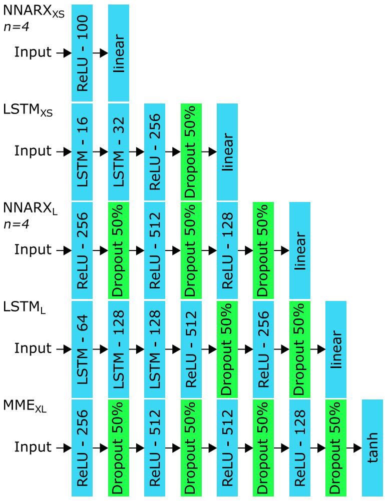

Fig. 3: Architectures of networks.

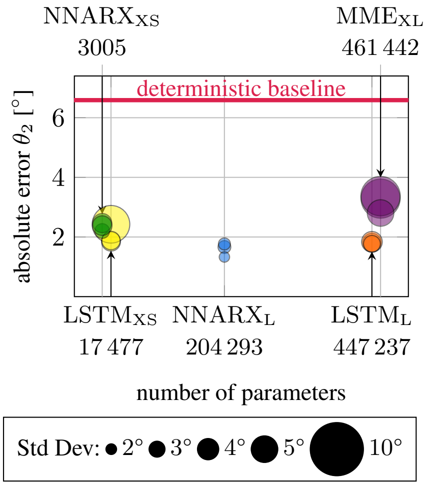

Fig. 4: Prediction of 100 time instants.

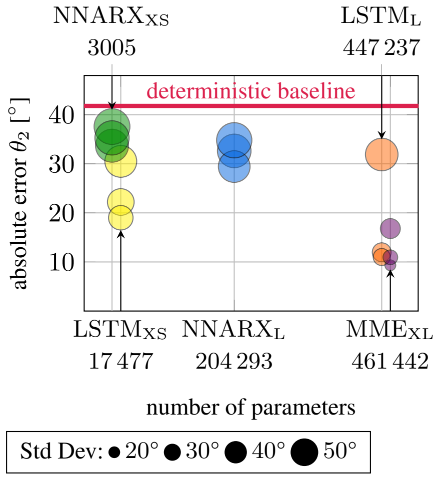

Fig. 5: Prediction of 300 time instants.

The generated training data consists of time series, such as swing-up, random walk, free fall and step response experiments. It is recorded at a frequency of 300 Hz. The measurements are mirrored around the pendulum angle $\theta_2 = 0°$ because the Furuta pendulum is symmetric around this axis. As a result, two hours and 20 minutes of data is generated, 80 % of which is used for training and 20 % for testing. The input of the NNARX and LSTM models is the state vector $\boldsymbol{x}$ from previous time steps and the torque τ. The output is the state vector $\boldsymbol{x}$ of the current time step. The MME model receives all states, except the arm angle θ_1, and the system input τ. The arm angle θ_1 is not required because the Furuta pendulum is rotational invariant around this axis. The output is the unmodeled dynamics on acceleration level, which is determined by the MME filter.

For each method, the number of hidden layers and neurons are varied. Then, for each model three independent trainings are performed. It is desired that these trained models have a similar low prediction error. Thus, an exceptionally good result obtained by chance is unlikely to be reproduced.

3 Results and Conclusion

The different neural networks are compared with respect to the error of the pendulum angle θ_2 because it is most relevant for control tasks. For each time series of 100 time instants, the maximum absolute error can be determined. The resulting error over all time series can then be averaged using the mean. Based on this evaluation, the best architectures for this test case are displayed in Fig. 3. The prediction error is presented in Fig. 4. Each circle represents a neural network and the size of the circle indicates the standard deviation (Std Dev) of the error. The error of the deterministic equation of motion is used as a baseline. All trained neural networks achieve a smaller prediction error than the deterministic model. The lowest prediction error is achieved by the $\mathrm{NNARX_L}$ model. The error of the $\mathrm{LSTM_L}$ model is slightly higher but has twice the number of parameters compared to the $\mathrm{NNARX_L}$ model. Even though LSTM models are far more complex and more expensive to train, the simple NNARX models are slightly better in this real world scenario. The $\mathrm{MME_{XL}}$ model performs the worst. A possible reason for this can be the noisy acceleration to be reproduced, which may result in prediction inaccuracies. In Fig. 5 the prediction horizon is increased to 300 time instants. In this case the $\mathrm{MME_{XL}}$ model performs best. The combination with the equation of motion seems to reproduce long term behavior well.

To measure the prediction time, the prediction horizon is reduced to 20 time instants. This is equivalent to 67 ms. The reduced prediction horizon represents a more reasonable test case in real-time applications due to computational limitations. The lowest prediction time is achieved by the simple $\mathrm{NNARX_{XS}}$ and $\mathrm{NNARX_L}$ model with 2.8 ms and 4.2 ms respectively. The more complex $\mathrm{LSTM_{XS}}$ requires 8.3 ms and the $\mathrm{LSTM_L}$ 12 ms. The $\mathrm{MME_{XL}}$ model takes the longest at 50 ms because the integrator requires four evaluations of the model in one time step and the equation of motion needs to be solved.

The results for the Furuta pendulum show that NNARX models are well suited for short-term predictions because the prediction time is low and the prediction error is small. If a long-term behavior needs to be modeled, LSTM and MME models are more appropriate.

Acknowledgements Open access funding enabled and organized by Projekt DEAL.

References

[1] Hochreiter, S.; Schmidhuber, J.: Long short-term memory. Neural Computation, Vol. 9, No. 8, pp. 1735–1780, 1997.
[2] Kolodziej, J.R.; Mook, D.J.: Model determination for nonlinear state-based system identification. Nonlinear Dynamics, Vol. 63, No. 4, pp. 735–753, 2010.
[3] Norgaard, M.; Poulsen, N.K.; Ravn, O.; Hansen, L.K.: Neural Networks for Modelling and Control of Dynamic Systems. Springer London, 2003.

