Research Seminar

Data-Driven Identification of Stable Non-Linear Systems Using Long Short-Term Memory

Student: Jakob Lerch

Supervisor: Irene Schimperna

Responsible Professors: Prof. Dr.-Ing. Patrick Mäder
Prof. Dr. Karl Worthmann

February 2025
TU Ilmenau

Abstract

This report addresses the topic of data-driven system identification. Its focus lies on how to identify a stable non-linear system using a long short-term memory (LSTM) network and how to analyze the retrieved model's stability properties in terms of input-to-state stability (ISS) and incremental input-to-state stability (δISS). Considering upper boundaries of the LSTM network's state, sufficient conditions for its ISS and δISS are derived. A method to enforce the model to be ISS and δISS during the identification procedure is shown. The results enable for guarantees regarding stability and boundedness and can be applied to controller design, *e.g.* with model predictive control. Future work could be to investigate similar methods for unstable non-linear systems.

1 Introduction

In Systems Engineering, a fundamental task is to retrieve a mathematical model from a system under consideration (which here is synonymous to *plant*). This act is referred to as *model retrieval* or *system identification*. Such a model can then be used for simulation and prediction of the system behavior, *e.g.* as a component in a control algorithm [1].

There are knowledge-based and data-driven approaches for model retrieval. Knowledge-based approaches model the process from domain knowledge, about the internal dynamics of the system [2]. What limits this approach is lack of understanding of the considered system's behavior as well as possibly missing information about relevant parameters. Data-driven approaches retrieve a model only from available data from the plant, such as corresponding in- and output time series [2]. Availability of large process data sets and the high popularity of

data-driven system identification in practice motivate investigating data-driven methods [3]. Methods that combine knowledge-based and data-driven techniques are referred to as *grey-box approaches* [4]. Thereby, a general model structure is imposed justified by domain knowledge, but several components of that model are tuned using available process data.

In data-driven system identification, the model to be fitted is a parameterization of a system's state and output equations. Finding parameters for the model to resemble the plant can be seen as a regression problem [2] where its parameters are fit to minimize a distance metric between the considered system and the model. This metric can be the mean squared error between the plant's and the model's output given the same input series, or the prediction error, which takes into account the model's capability of predicting future time steps given corresponding input data, but only the present time step's state [2]. If the considered system is assumed to be linear, the class of considered models for identification are different variants of parameterizing linear discrete state equations, such as the Box-Jenkins model or the auto-regressive model exogenous (ARX) model [2].

In the case of non-linear systems, considered model classes include *e.g.* the non-linear ARX (NARX) model. The NARX model is a linear combination of non-linear basis functions where parameters like the linear factors and argument offsets of basis functions are fitted [2]. Another model class is that of recurrent neural networks (RNNs). Their high representational power and wide applicability [4] motivates their investigation. In [4] various and in general well-known RNN architectures are proposed for the use of non-linear system identification, including neural nonlinear ARX (NNARX), echo state networks (ESN), and gated RNNs, namely long short-term memory (LSTM) networks and gated recurrent units (GRUs). Our interest is in LSTM networks in particular.

Following [4] one important question concerning RNNs for the use in control systems is that of robustness and stability. In [3] this topic is addressed and sufficient conditions for input-to-state stability (ISS) and incremental input-to-state stability δISS of LSTM networks as well as a method to integrate those conditions in the network's optimization procedure are proposed.

An application of RNN-models where stability can be guaranteed is their use in model predictive control (MPC) algorithms. In [3], an MPC algorithm for correctly tracking a sequence of constant reference points is proposed. This involves constructing an observer based on the LSTM network resembling the plant which is assumed to be stable itself. In [5] this approach is being developed further to also be offset-free in case of output disturbances. This is accomplished by introducing a disturbance model in the LSTM-based observer [5]. In [6] an MPC algorithm is proposed which does not require any assumptions regarding the plant's stability.

This report considers an LSTM network identified from a plant's available data. It mainly addresses two questions: first, if and how the trained network can be analyzed to say if it is stable or not and second, if there is a way to force the LSTM to be stable during the identification process given a stable plant.

2 Input-to-State Stability for LSTM Networks

We are considering the system Σ, which is defined as follows.

$$\tilde{x}^+ = f(\tilde{x}, u) \tag{1}$$
$$\tilde{y} = h(\tilde{x}), \tag{2}$$

using only data of the input u and output $\hat{y}$. Thereby, Σ is stable, non-linear, time-invariant, and its input u is bounded: $||u||_\infty \in [u_{min}, u_{max}]$.

To specify the exact meaning of *stability*, input-to-state stability (ISS) and incremental input-to-state stability (δISS) are defined below. They provide a notion for stability with respect to an input signal being present. To be able to define ISS and δISS, comparison functions need to be defined first. The definitions are as follows [7].

Definition 1 (Comparison functions)**.**

$$\begin{aligned}
\mathcal{K} &:= \{\alpha : \mathbb{R}^+ \to \mathbb{R}^+ \quad &| \ \alpha \textit{ continuous, strictly increasing, } \alpha(0) = 0\} \\
\mathcal{K}_\infty &:= \{\alpha : \mathbb{R}^+ \to \mathbb{R}^+ \quad &| \ \alpha \in \mathcal{K}, \ \alpha \textit{ unbounded}\} \\
\mathcal{KL} &:= \{\beta : \mathbb{R}^+ \times \mathbb{N}_0 \to \mathbb{R}^+ \quad &| \ \beta \textit{ cont., } \beta(\cdot, k) \in \mathcal{K}, \\
& & \quad \beta(s, \cdot) \textit{ strictly decreasing, } \lim_{k\to\infty} \beta(s, k) = 0\}
\end{aligned}$$

Definition 2 (Input-to-state stability (ISS))**.** *System Σ is called* input-to-state stable *in a state space $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ with respect to an input space $\mathcal{U} \subseteq \mathbb{R}^{n_u}$, if there exist functions $\beta \in \mathcal{KL}$ and $\gamma_\nu \in \mathcal{K}_\infty$ such that, for any $k \in \mathbb{N}_0$, any initial state $x_0 \in \mathcal{X}$, any input sequence $(\nu(h))_{h\in\mathbb{N}_0}$, it holds that:*

$$||x(k)|| \leq \beta(||x_{01}||, k) + \gamma_\nu(\max_{h\geq 0} ||\nu(h)||). \tag{3}$$

where $||\cdot||$ denotes the L_2-norm and n_a denotes the dimension of an arbitrary vector a.

Intuitively, if a system is ISS, its state gets near an equilibrium point after some transient phase. Its distance from the equilibrium point is then bounded only by a function of the input series.

Definition 3 (Incremental input-to-state stability (δISS))**.** *System Σ is called* incrementally input-to-state stable *in a state space $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ with respect to an input space $\mathcal{U} \subseteq \mathbb{R}^{n_u}$, if there exist functions $\beta_\delta \in \mathcal{KL}$ and $\gamma_\delta \in \mathcal{K}_\infty$ such that, for any $k \in \mathbb{N}_0$, any initial states $x_{01}, x_{02} \in \mathcal{X}$, and any input sequences $((\nu_1(h))_{h\in\mathbb{N}_0}, (\nu_2(h))_{h\in\mathbb{N}_0})$, it holds that*

$$||x_1(k) - x_2(k)|| \leq \beta_\delta(||x_{01} - x_{02}||, k) + \gamma_\delta(\max_{h\geq 0} ||\nu_1(h) - \nu_2(h)||). \tag{4}$$

With δISS, the distance between the state trajectories evolving from two different values of the initial state is considered. Intuitively, if a system is δISS, the two trajectories eventually get near each other. Their distance is then only bounded by a function of the distance of the input series for the respective scenarios. Note that ISS is a special case of δISS where one of the state trajectories is equal to 0, thus δISS implies ISS.

The objective is now to find a model for the system Σ. In [3] this is done using a long short-term memory (LSTM) network, which is defined below.

Definition 4 (Long short-term memory (LSTM))**.** *The LSTM network is the following set of equations*

$$\hat{x}^+ = \sigma_g(W_f u + U_f \xi + b_f) \circ \hat{x} + \sigma_g(W_i u + U_i \xi + b_i) \circ \sigma_c(W_c u + U_c \xi + b_c), \tag{5}$$

$$\xi^+ = \sigma_g(W_o u + U_o \xi + b_o) \circ \sigma_c(\hat{x}^+), \tag{6}$$

$$y = C\xi + b_y. \tag{7}$$

with an input $u \in \mathbb{R}^{n_u}$*, an output* $y \in \mathbb{R}^{n_y}$*, a* hidden state $x \in \mathbb{R}^{n_{\hat{x}}}$*, an* output state $\xi \in \mathbb{R}^{n_{\hat{x}}}$*, an (overall) state* $x := (\hat{x},\ \xi)^T$*,* weight matrices $W_f, W_i, W_o, W_c \in \mathbb{R}^{n_{\hat{x}} \times n_u}, U_f, U_i, U_o, U_c \in \mathbb{R}^{n_{\hat{x}} \times n_{\hat{x}}}, C \in \mathbb{R}^{n_y \times n_{\hat{x}}}$ *and* bias vectors $b_f, b_i, b_o, b_c \in \mathbb{R}^{n_{\hat{x}}}$ *and* $b_y \in \mathbb{R}^{n_y}$*. The* activation functions $\sigma_g(x)$ *and* $\sigma_c(x)$ *are nonlinear functions with* $\sigma_g(x) = \frac{1}{1+e^{-x}}$ *and* $\sigma_c(x) = tanh(x)$*. The operation* $\circ$ *is the element-wise product.*

The parameters of the network, namely the weight matrices and the bias vectors, are chosen to minimize cost function. In [3] this is the mean squared error (MSE) between the plant's output $\hat{y}$ and the network's output y given a common input sequence. This is usually done using an iterative, non-linear optimization algorithm, such as RMSProp. In [2] it is pointed out that using a prediction error is possible as well. It could be examined further if this would have any desirable effect on the training procedure or on the network's performance.

When identifying a model from a plant, it is desirable for the model to behave as similar to the plant as possible. Therefore, since the plant is stable, we require the LSTM network to be stable as well. More precisely, we require that the network is ISS if the plant is ISS and that the network is also δISS if the plant is δISS.

However, there is a special case of an LSTM network's configuration where it does not fulfill the inequation in Definition 2 anymore, but its state is still bounded. Take Definition 4 and assume $\hat{x} = \xi = 0$ and for all time steps the input u 0 as well. Then, to fulfill Definition 2, x^+ would need to be 0 as well. However, following Definition 4, it would be a function of bias terms:

$$\hat{x}^+ = \sigma_g(b_i) \circ \sigma_c(b_c). \tag{8}$$

Therefore, in [3] an alternative definition for ISS is used, which is as follows.

Definition 5 (Input-to-state stability (ISS) for LSTM networks)**.** *System* Σ *is called* input-to-state stable *in* $\mathcal{X}$ *with respect to* $\mathcal{U}$*, if there exist functions* $\beta \in \mathcal{KL}$ *and* $\gamma_\nu, \gamma_b \in \mathcal{K}_\infty$ *such that, for any* $k \in \mathbb{N}_0$*, any initial state* $x_0 \in \mathcal{X}$*, any input sequence* $(\nu(h))_{h \in \mathbb{N}_0}$*, and any bias* $b_c \in \mathbb{R}$*, it holds that:*

$$||x(k)|| \leq \beta(||x_0||, k) + \gamma_\nu(\max_{h \geq 0} ||\nu(h)||) + \gamma_b(||b_c||). \tag{9}$$

Apart from the above motivation, an LSTM network being ISS or δISS has some other advantages. At the start of inferring an output series from it, the state has to be initialized. This is usually done randomly, without any prior knowledge about the plant itself or its state. If an LSTM network is ISS, its state eventually is bounded only by the input and a constant bias. This is true regardless the value of x_0. If an LSTM network is δISS, the distance of two trajectories eventually gets equal to each other given the same input series. Therefore, the state initialization does not affect predictions on the long term. Other than that, ISS enables for guarantees regarding the bounds of state and output in practical situations which is relevant *e.g.* when considering safety.

This motivates the question, if it is possible to guarantee ISS or δISS for an LSTM network formally. In [3] Theorems 1 and 2 are derived that give sufficient conditions for an LSTM network to be ISS or δISS.

Theorem 1. *The LSTM network is ISS with respect to the input u and bias b_c if A is Schur, where*

$$A = \begin{bmatrix} \overline{\sigma}_g^f & \overline{\sigma}_g^i ||U_c|| \\ \overline{\sigma}_g^o \overline{\sigma}_g^f & \overline{\sigma}_g^o \overline{\sigma}_g^i ||U_c|| \end{bmatrix}. \tag{10}$$

Theorem 2. *The LSTM network is δISS with respect to the inputs u_1 and u_2 if A_δ is Schur, where*

$$A_\delta = \begin{bmatrix} \overline{\sigma}_g^f & \alpha \\ \overline{\sigma}_g^o \overline{\sigma}_g^f & \alpha \overline{\sigma}_g^o + \frac{1}{4} \overline{\sigma}_c^x ||U_o|| \end{bmatrix} \tag{11}$$

with

$$\alpha = \frac{1}{4} ||U_f|| \frac{\overline{\sigma}_g^i \overline{\sigma}_c^c}{1 - \overline{\sigma}_g^f} + \overline{\sigma}_g^i ||U_c|| + \frac{1}{4} ||U_i|| \overline{\sigma}_c^c. \tag{12}$$

The symbols $\overline{\sigma}_g^o$, $\overline{\sigma}_g^i$, $\overline{\sigma}_c^c$, $\overline{\sigma}_c^x$ denote upper bounds for the corresponding activation functions in Definition 4. Theorem 1 is proven by for each of the state equations in Definition 4 taking the norm on both sides, and finding upper bounds of its components to get an inequation of the form

$$\begin{bmatrix} ||\hat{x}^+|| \\ ||\xi^+|| \end{bmatrix} \leq g(\begin{bmatrix} ||\hat{x}|| \\ ||\xi|| \end{bmatrix}, ||u||, ||b_c||). \tag{13}$$

where g resembles the upper bound of the right-hand side of the ininequation. The inequation can be reformulated to be

$$||x^+|| \leq A||x|| + B_u||u|| + B_b||b_c||. \tag{14}$$

with $A \in \mathbb{R}^{2\times 2}, B_u, B_b \in \mathbb{R}^{2\times 1}$. Iterating ((14)) and again taking the norm on both sides leads to an inequation of the same form as in Definition 5 with $\beta(||x_0||, k) = ||A^k||\ ||x_0||$. Hence, if A is Schur, $\beta \in \mathcal{KL}$ and therefore, the LSTM network is ISS. Theorem 2 is proven in a similar way, but considering the distances of two state trajectories instead of one state trajectory itself.

Using Lemma 1, Propositions 1 and 2 are derived that give easy-to-check criteria for the conditions in Theorems 1 and 2.

Lemma 1. *Given a 2 × 2 real matrix A, it is Schur if and only if*

$$-1 - a < b < 1, \tag{15}$$

where $a = -A_{11} - A_{22}$ and $b = A_{11}A_{22} - A_{12}A_{21}$.

Proposition 1. *A is Schur if and only if the following inequation holds:*

$$\overline{\sigma}_g^f + \overline{\sigma}_g^o \overline{\sigma}_g^i ||U_c|| < 1. \tag{16}$$

Proposition 2. *A_δ is Schur if and only if the following inequation hold:*

$$-1 + \overline{\sigma}_g^f + \alpha \overline{\sigma}_g^o + \frac{1}{4} \overline{\sigma}_c^x ||U_o|| \ < \ \frac{1}{4} \overline{\sigma}_g^f \overline{\sigma}_c^x ||U_o|| \ < \ 1. \tag{17}$$

3 Enforcing ISS or δISS during Identification

The inequations in Propositions 1 and 2 allow for checking stability of an LSTM network after completing the identification procedure. For a possible application this could mean repeating the identification multiple times until the desired criteria are fulfilled. It would be much more practical, if there was a strategy to force the parameters of the network during training towards fulfilling the desired stability criteria. In [3] this is done by augmenting the loss function. Here, the procedure is explained only for enforcing ISS. However, enforcing δISSrequires just the same steps.

Without augmentation, a standard loss function for training the network looks as follows

$$L(W,b) = \frac{1}{N}\sum_{k=0}^{N-1} ||\tilde{y}(k) - y(k)||_2^2, \tag{18}$$

where $\tilde{y}$ is the output of the original system Σ (see (1)) and y is the output of the LSTM network (see Definition 4). To incorporate the inequation from 1 in it, it is reformulated so that the right-hand side is zero:

$$\overline{\sigma}_g^f + \overline{\sigma}_g^o \overline{\sigma}_g^i ||U_c|| - 1 < 0. \tag{19}$$

The term on the right-hand side is called the *residual term* r. To enforce $r < 0$, the loss function can be augmented to be

$$\begin{aligned} L(W,b) = &\frac{1}{N}\sum_{k=0}^{N-1} ||\tilde{y}(k) - y(k)||_2^2 + \\ &+ \rho\ r. \end{aligned} \tag{20}$$

where $\rho \in \mathbb{R}$ is a hyper-parameter to be chosen.

The authors of [3], however, choose to augment the loss function with a piece-wise linear function in r to avoid too large fulfillment of the ISS criterion while being less optimal considering the MSE:

$$\begin{aligned} L(W,b) = &\frac{1}{N}\sum_{k=0}^{N-1} ||\tilde{y}(k) - y(k)||_2^2 + \\ &+ (\rho^- \min(r,0) + \rho^+ \max(r,0)) \end{aligned} \tag{21}$$

where $\rho^+, \rho^- \in \mathbb{R}$ are hyper-parameters to be chosen. Additionally, the authors report rather small values for ρ^+ (magnitude of 10^{-3}) and ρ^- (magnitude of 10^{-5}).

For enforcing δISS, one needs to consider the inequations from Proposition 2 instead. The procedure, however, is the same as above. Note that Proposition 2 contains two inequations and thus would yield two residual terms.

4 Conclusion

This report addressed the stability analysis of LSTMs for their use in system identification of stable non-linear systems. Several stability definitions for non-linear systems were investigated. Formal conditions for ISS/δISS of an LSTM network were reported and a method for enforcing the ISS/δISS property of an LSTM during training given an ISS/δISS plant was proposed.

Further research could include stability analysis of different model types, data-driven system identification and control of unstable systems (where [6] could be a good starting point), probabilistic methods for system identification accounting for model uncertainty and incremental training techniques, as suggested in [4].

References

[1] F. Iacono, J. L. Presti, I. Schimperna, et al., "Improvement of manufacturing technologies through a modelling approach: An air-steam sterilization case-study," *Procedia Computer Science*, Proceedings of the 2nd International Conference on Industry 4.0 and Smart Manufacturing (ISM 2020), vol. 180, pp. 162–171, Jan. 1, 2021, ISSN: 1877-0509. DOI: 10.1016/j.procs.2021.01.357. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921004336 (visited on 02/11/2025).

[2] Y. A. W. Shardt and H. Weiß, "Modellierung dynamischer Prozesse mit Methoden zur Systemidentifikation," in *Methoden der Statistik und Prozessanalyse: Eine anwendungsorientierte Einführung*, Y. Shardt and H. Weiß, Eds., Berlin, Heidelberg: Springer, 2021, pp. 313–373, ISBN: 978-3-662-61626-0. DOI: 10.1007/978-3-662-61626-0_6. [Online]. Available: https://doi.org/10.1007/978-3-662-61626-0_6 (visited on 02/11/2025).

[3] E. Terzi, F. Bonassi, M. Farina, and R. Scattolini, "Learning model predictive control with long short-term memory networks," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 18, pp. 8877–8896, Dec. 2021, ISSN: 1049-8923, 1099-1239. DOI: 10.1002/rnc.5519. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/rnc.5519 (visited on 12/12/2024).

[4] F. Bonassi, M. Farina, J. Xie, and R. Scattolini, "On recurrent neural networks for learning-based control: Recent results and ideas for future developments," *Journal of Process Control*, vol. 114, pp. 92–104, Jun. 1, 2022, ISSN: 0959-1524. DOI: 10.1016/j.jprocont.2022.04.011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959152422000610 (visited on 11/08/2024).

[5] I. Schimperna, C. Toffanin, and L. Magni, "On offset-free model predictive control with long short-term memory networks," *IFAC-PapersOnLine*, 12th IFAC Symposium on Nonlinear Control Systems NOLCOS 2022, vol. 56, no. 1, pp. 156–161, Jan. 1, 2023, ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2023.02.027. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S240589632300215X (visited on 11/08/2024).

[6] D. Ravasio, M. Farina, and A. Ballarino, "LMI-based design of a robust model predictive controller for a class of recurrent neural networks with guaranteed properties," *IEEE Control Systems Letters*, vol. 8, pp. 1126–1131, 2024, Conference Name: IEEE Control Systems Letters, ISSN: 2475-1456. DOI: 10.1109/LCSYS.2024.3408040. [Online]. Available: https://ieeexplore.ieee.org/document/10543173 (visited on 02/11/2025).

[7] L. Grüne and J. Pannek, "Discrete time and sampled data systems," in *Nonlinear Model Predictive Control: Theory and Algorithms*, L. Grüne and J. Pannek, Eds., Cham: Springer International Publishing, 2017, pp. 13–43, ISBN: 978-3-319-46024-6. DOI: 10.1007/978-3-319-46024-6_2. [Online]. Available: https://doi.org/10.1007/978-3-319-46024-6_2 (visited on 02/11/2025).