

Space-Filling Regularization for Robust and Interpretable Nonlinear State Space Models

Hermann Klein^a, Max Heinz Herkersdorf^b and Oliver Nelles^c

University of Siegen, Department Mechanical Engineering, Automatic Control – Mechatronics,
Paul-Bonatz-Str. 9-11, 57068 Siegen, Germany

Keywords: Nonlinear System Identification, State Space Models, Regularization, Local Model State Space Network, Space-Filling.

Abstract: The state space dynamics representation is the most general approach for nonlinear systems and often chosen for system identification. During training, the state trajectory can deform significantly leading to poor data coverage of the state space. This can cause significant issues for space-oriented training algorithms which e.g. rely on grid structures, tree partitioning, or similar. Besides hindering training, significant state trajectory deformations also deteriorate interpretability and robustness properties. This paper proposes a new type of space-filling regularization that ensures a favorable data distribution in state space via introducing a data-distribution-based penalty. This method is demonstrated in local model network architectures where good interpretability is a major concern. The proposed approach integrates ideas from modeling and design of experiments for state space structures. This is why we present two regularization techniques for the data point distributions of the state trajectories for local affine state space models. Beyond that, we demonstrate the results on a widely known system identification benchmark.

1 INTRODUCTION

Models of dynamic systems are the base for numerous industrial applications, especially for control design or virtual sensing. Insofar as white box models derived from physical principles oftentimes fail to meet an acceptable tradeoff between modeling effort and accuracy, data-driven model learning, also known as system identification, comes into play. Here, the user can choose between different mathematical architectures to describe the system's dynamics. A state space representation is the most abstract and general kind as well as required for many control design methods. Examples of alternative approaches are feedforward-trained models like finite impulse response or nonlinear autoregressive models with exogenous input or output feedback-trained kinds like output error models (Ljung, 1999). In comparison to the latter and especially for the nonlinear case, the state space model is a more powerful modeling approach due to its internal state feedback (Belz et al., 2017; Schüssler et al., 2019).

Whereas *linear* state space models can be extracted from frequency domain approaches with subspace identification (Pintelon and Schoukens, 2012; Van Overschree and De Moor, 1995), *nonlinear* models have to be trained via nonlinear optimization with the objective of simulation error fit. They generally arise when the linear state and output functions are replaced with nonlinear function approximators. Here, various approaches exist, including polynomials (Paduart et al., 2010) or neural networks (Suykens et al., 1995; Forgione and Piga, 2021). Due to local interpretability and explainability reasons, we focus on the class of local affine state space models. Mathematically related architectures are (softly switching) piecewise affine (Garulli et al., 2012), Takagi Sugeno (TS) or quasi-Linear Parameter Varying (qLPV) state space models (Rotondo et al., 2015).

In general, system identification tries to find an accurate model based on input/output training datasets (Ljung et al., 2020). As a consequence, an immanent feature of nonlinear state space models is the implicit optimization of the model's state, since it depends on trainable parameters. Due to the simulation error fit, the state is optimized indirectly. In geometrical terms, deformations of the trajectories can

^a <https://orcid.org/0009-0009-5386-3575>

^b <https://orcid.org/0009-0007-8554-8153>

^c <https://orcid.org/0000-0002-9471-8106>

be observed within the optimization procedure. This can be visualized in the phase plot of the state variables' trajectories, at least for the two-dimensional projection¹. The point distribution in the state space is manipulated until the termination of the optimization. For TS models, the effects on the state trajectory can be connected with local regions. This interaction can affect robustness and interpretability and is addressed in this contribution.

As a first try, any system identification task can be challenged with a pure data-driven (black box) approach. In such a setting, the state space model is fully parametrized including redundant parameters. A common technique in estimation theory is to use regularization by application of an additional penalty term to the objective function (Boyd and Vandenberghe, 2004). In the context of nonlinear state space modeling, recent work deals with L1 regularization for black box models (Bemporad, 2024). Moreover, Liu et al. (2024) use regularization to make a black-box state meet a physical-interpretable state realization. An overview on methods for nonlinear systems can e.g. be found in Pillonetto et al. (2022).

As mentioned above, the interaction of the data point distribution with local activation functions is crucial for models with regional organization, like the Local Model State Space Network (LMSSN). As a consequence, we develop a *space-filling* regularization for the LMSSN procedure. This transfers ideas from the Design of Experiments into model training. The novelty of this contribution is the development of a space-filling penalty term. We analyze it in detail and compare two related regularization techniques.

The article is organized as follows. In Section 2, we present the LMSSN-based modeling procedure. Next, we present the evolution of the space-filling indicators within the optimization procedure in Section 3. The effect of a space-filling penalty term is presented in Section 4. Our method is tested on a benchmark process in Section 5. Finally, we summarize our work and give an outlook on future lines of research.

2 NONLINEAR STATE SPACE MODELING

The proposed method is suitable for all state space models in which the state explores an a priori known range. Since we focus on the LMSSN method, its functionality is introduced in this section. This un-

regularized version serves as a reference for recent advancements regarding space-filling regularization.

2.1 Local Model State Space Network

LMSSN is a system identification method to create discrete-time nonlinear state space models. The mathematical model architecture embeds Local Model Networks (LMNs) in a state space framework. LMSSN modeling results in local affine state space models, weighted with normalized radial basis functions (NRBF). Detailed information can be found in Schüssler (2022). It is emphasized that LMSSN differs from TS and qLPV approaches in the use of the Local Linear Model Tree (LOLIMOT). The tree-construction algorithm serves for the determination of the fuzzy premise variables (or scheduling variables in qLPV terms). Additionally, it uses a heuristic axis-orthogonal input space partitioning approach by automatic parametrization of the validity functions. For more information about LOLIMOT, we refer to Nelles (2020).

The resulting LMSSN model is formulated with the following equations,

$$\begin{aligned}\hat{\mathbf{x}}(k+1) &= \sum_{j=1}^{n_{\text{LM},x}} (\underline{A}_j \hat{\mathbf{x}}(k) + \underline{b}_j u(k) + \underline{o}_j) \circ \Phi_j^{[x]}(k) \\ \hat{\mathbf{y}}(k) &= \sum_{j=1}^{n_{\text{LM},y}} (\underline{c}_j^T \hat{\mathbf{x}}(k) + d_j u(k) + p_j) \circ \Phi_j^{[y]}(k).\end{aligned}\quad (1)$$

For the single-input single-output case ($n_u=1$ input and $n_y=1$ output variable), the tensors $\underline{A}_j \in \mathbb{R}^{n_x \times n_x}$, $\underline{b}_j \in \mathbb{R}^{n_x}$, $\underline{o}_j \in \mathbb{R}^{n_x}$, $\underline{c}_j \in \mathbb{R}^{n_x}$, $d_j \in \mathbb{R}$ and $p_j \in \mathbb{R}$ are filled with the slopes and offsets of the j -th local affine model (LM). The dynamical order is expressed by the number of state variables n_x and the discrete-time is indicated with k . In configuration (1), there are altogether $n_{\text{LM},x}$ superposed affine models for the state $\hat{\mathbf{x}}(k+1)$ and $n_{\text{LM},y}$ for the output $\hat{\mathbf{y}}(k)$. The basis functions $\Phi_j^{[x]}$ and $\Phi_j^{[y]}$ express the j -th local validity function. Since they are implemented with normalized Gaussians, the partition of unity holds, $\sum_{j=1}^{n_{\text{LM},x}} \Phi_j^{[x]} = 1$ and $\sum_{j=1}^{n_{\text{LM},y}} \Phi_j^{[y]} = 1$.

Due to the design of LMSSN in state space form, the parameter initialization can be done deterministically with a global linear state space model. The latter is gathered as the Best Linear Approximation (BLA) of the process in the frequency domain (McKelvey et al., 1996). Next, subspace-based identification converts the *nonparametric* frequency response function model into a *parametric* linear state space model. The fully-parametrized discrete-time model is then transformed into a balanced realization via a sim-

¹For simplicity, the phase plot of the state variables is designated as the *state trajectory* in this paper.

ilarity transformation (Verriest and Kailath, 1983; Lunenberger, 1967). It evokes a diagonal structure within the state space matrices which leads to a state realization with an evenly (and thus desired) trajectory shape.

After the linear model initialization, the LOLIMOT algorithm starts. With the help of the stepwise addition of weighted LMs, it enables for nonlinear curve approximation. Each LM is connected with a region of activation in the extended input/state space $\tilde{u} = [\hat{x}, u]^T$. This corresponds to the partition of $\tilde{u}$ with interpolation regimes for smooth cross-fading in between. It can be seen as a generation of *splits* of $\tilde{u}$, arranged by center placement of the NRBFs. If every dimension of $\tilde{u}$ is to be analyzed in form of a split, $n_{\text{split}} = \dim(\tilde{u})$ split options arise. For the ease of the center placement, it is intuitive to scale the state trajectory into a fixed interval, preferably the unit cube. The scaling is actualized with an affine scaling transformation, parametrized with range and offset values according to minimum and maximum of $\hat{x}$. The procedure is called *split-adaption algorithm* according to Schüssler et al. (2019). Each LOLIMOT iteration investigates all n_{split} options for the region with the largest error separately via an optimization run.²

2.2 Problem Statement

The loss function J being minimized in each LOLIMOT iteration is the sum of the squared output errors. For a one-dimensional output variable, the optimization problem is stated as

$$\begin{aligned} \min_{\theta} J(\theta) &= \min_{\theta} \sum_{k=1}^N \left(y(k) - \hat{y}(k; \theta) \right)^2 \\ \text{s.t. } \hat{x}(k+1) &= \text{LMN}(\hat{x}(k), u(k); \theta^{[x]}) \\ \hat{y}(k) &= \text{LMN}(\hat{x}(k), u(k); \theta^{[y]}), \end{aligned} \quad (2)$$

where θ is the vector of all model parameters and N is the number of samples collected for training. The nonlinear problem is solved with a Quasi-Newton optimizer with BFGS Hessian approximation. The optimization is run till convergence in a local minimum terminating on a loss and gradient tolerance or minimal step size.

By fitting the parameters according to the minimization of the output error, the state trajectory changes its shape from the balanced realization to the one that is most effective for the objective (2). Next, we show a possible consequence on the state

²Note the LOLIMOT iterations have to be distinguished from the gradient-based iterations within the nonlinear optimization of each split option.

trajectory for the first split of a simulated second-order single-input single-output system with nonlinear feedback,

$$\ddot{y}(t) + a_1 \dot{y}(t) + a_0 f(y(t)) = b_0 u(t), \quad (3)$$

with the parameter values $a_0=b_0=15$ and $a_1=3$. The nonlinear feedback $f(y)$ is arranged with a smooth exponentially rising curve. Figure 1 demonstrates that the initial state trajectory was within the unit cube, whereas the optimized one is compressed. Obviously, the nonlinear behavior of the process can be captured although no data point is fully active in the upper LM. Nevertheless, LM2 influences the data points via smooth cross-fading of both LMs in the interpolation regime. The shown behavior causes problems regarding interpretation. Furthermore, the effect is accompanied by robustness problems. Large local parameter values of LM2 are necessary to influence data, which are located in the lower LM1. With the help of the maximum absolute value of the eigenvalues μ_i of the local state transition matrices,

$$\underline{A}_{\text{LM},j} = \begin{bmatrix} \underline{a}_{1,j}^\top \\ \underline{a}_{2,j}^\top \end{bmatrix},$$

unstable poles in LM2 can be detected:

$$\begin{aligned} \max_i |\mu_i(\underline{A}_{\text{LM},1})| &= 0.99 < 1, \\ \max_i |\mu_i(\underline{A}_{\text{LM},2})| &= 4.23 > 1 \end{aligned}$$

The phenomenon can be seen as overfitting and contains the risk of unstable extrapolation behavior. Finally, the risk of reactivation (Nelles, 2020) of LMs within the next splits is increased, since the split adaption leads to major differences in the NRBF's standard deviations.

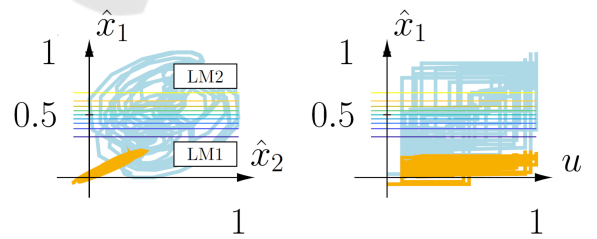


Figure 1: Extended state trajectory before (●) and after (●) the optimization of the first split in the $\hat{x}_1$ -dimension. The colored contour lines show the validity functions and mark the interpolation regime. For brevity, the $\hat{x}_2$ - u projection is left out.

Note that the fixed centers of the LOLIMOT-partitioned state space make this effect visible. It can hardly be investigated for deep neural state space models or recurrent neural networks, where all neural weights are trainable parameters. Furthermore, it

must be mentioned that the observed behavior (Fig. 1) enables modeling with low flexibility. In the case of LMSSN this means that only one split is sufficient for the complete identification. Thus it is very effective because only two local *linear* models are able to fit a process with a smooth curve (3).

3 SPACE-FILLING EVOLUTION

Before merging modeling with space-filling properties, three suitable metrics for the assessment of data point distributions are presented. Afterwards, their evolution along the optimization progress is demonstrated. For mathematical reasons, the state trajectory is described as the set of points in the extended input/state space as

$$\mathcal{X} = \{\tilde{\mathbf{u}}(k), \forall k \in \{1, \dots, N\}\}.$$

3.1 Metrics and Indicators

Inspired by the shape of the state trajectory in Fig. 1, the covered volume is an intuitive and straightforward space-filling indicator. It can be assessed as the volume of a convex hull on the state trajectory (CHV) (Boyd and Vandenberghe, 2004). Written for the extended input/state space, the volume V is defined as

$$V = \frac{1}{\dim(\tilde{\mathbf{u}})} \sqrt{\det(\underline{G})}, \quad (4)$$

where the Gram matrix $\underline{G}$ is filled with

$$G_{i,j} = \tilde{\mathbf{u}}(i)^T \tilde{\mathbf{u}}(j), \quad \forall i, j \in \mathcal{H}.$$

The convex hull is defined by the set of points $\mathcal{H} \subseteq \{1, 2, \dots, N\}$ derived with the help of the quickhull algorithm. Intuitively, the CHV is going to deliver large values for widely distributed points and small values for concentrated points.

Alternatively, support point-based indicators are commonly applied for assessing space-filling measurements. Here, the state trajectory is compared to a given distribution of n_g grid or Sobol points $\mathcal{G}$. In this contribution, we concentrate on a uniform grid, implying that a uniform distribution is desired. This leads to the final two metrics.

First, the mean of the minimum distances ψ_p from every grid point $\mathbf{g}_j, \forall j \in \{1, 2, \dots, n_g\}$, to the nearest extended input/state space point $\tilde{\mathbf{u}}(k)$ is calculated (Herkersdorf and Nelles, 2025). The criterion is strongly related to Monte Carlo Uniform Sampling Distribution Approximation (Smits and Nelles, 2024).

In this paper, it is denoted with ψ_p and given as

$$\psi_p = \frac{1}{N} \sum_{j=1}^M \min_{k \in \{1, \dots, N\}} d(\tilde{\mathbf{u}}(k), \mathbf{g}_j) \quad (5)$$

with the Euclidian distance d

$$d(\tilde{\mathbf{u}}(k), \mathbf{g}_j) = \sqrt{(\tilde{\mathbf{u}}(k) - \mathbf{g}_j)^T (\tilde{\mathbf{u}}(k) - \mathbf{g}_j)}.$$

It is important to note that distance-based criteria such as ψ_p can encounter difficulties in high-dimensional spaces. Furthermore, its grid-oriented nature is affected by the curse of dimensionality. Both aspects are beyond the scope of this work, as we focus on lower-dimensional problems. Nonetheless, they constitute limitations of the approach.

Secondly, from a statistical point of view, the Kullback-Leibler Divergence (KLD) is able to assess the similarity of two probability density functions (PDF) Kullback and Leibler (1951). The PDF of a dataset $\mathcal{S}$ is denoted by $P(\mathcal{S})$. In the prevalent application, it compares the estimated probability density $\hat{P}(\mathcal{X})$ with the uniform distribution of a grid $P(\mathcal{G}) = 1/n_g$. The KLD is then calculated as

$$\text{KLD} = -\frac{1}{n_g} \sum_{k=1}^N \log(\hat{P}(\mathcal{X})). \quad (6)$$

The required probability density estimation of the state trajectory is done with kernel density estimation (Scott, 1992). For the measurement of the space-filling properties of a point distribution, both KLD and ψ_p will decrease when the state trajectory becomes uniformly, i.e., meets the grid.

3.2 Tracking Within Nonlinear Optimization

For the sake of plausibility, the previously introduced indicators in the example depicted in Fig. 1 are examined. Obviously, CHV decreases as the optimization progresses. On the other hand, KLD and ψ_p are expected to grow, because the state trajectory is evenly distributed *before* the optimization, and narrow-shaped *after* the optimization has converged. Figure 2 shows the evolution of the three space-filling indicators as well as the loss along with the iterations.

It can be stated that all indicators detect the shrinkage of the state trajectory and thus are validated. Note that the absolute numerical values of the indicators are not of interest here. The focus is only on the change during the optimization. This is why the curves are normalized w.r.t. their maximum values.

On closer inspection, it can be observed that the changing curvature reveals large rates around iteration

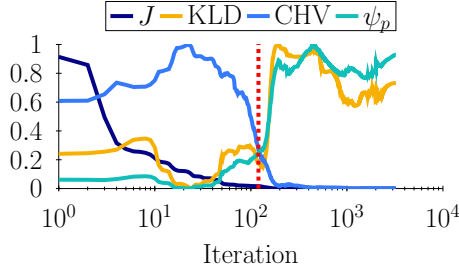


Figure 2: Tracking of loss and space-filling indicators according to the state trajectories depicted in Fig. 1.

110, indicated by the vertical line. Loosely spoken, the optimization has already reached an acceptable fit of the process output at this point which is visible in the loss curve. It can be concluded that space-filling measured by those indices gets worse *after* the model performance met a satisfying level. This confirms that the phenomenon of state trajectory deformation can be seen as an overfitting of nonlinear state space models. This is why we recommend a regularization technique with an early-stopping effect in the next section.

4 REGULARIZED IDENTIFICATION

Since strong state trajectory deformation is an unsatisfying effect, it is reasonable to design an additional penalty for the objective that drives its data points to a desired uniform distribution. In this section, we present two penalty term approaches for space-filling regularization of problem (2). Both approaches use the ψ_p described by (5) which was chosen due to implementation reasons. Generally, Fig. 2 suggests that all space-filling indicators would lead to similar results.

4.1 Space-Filling Penalty Term

As a consequence of the ψ_p indicator delivering low numerical values when data are space-filling and high values when data are non-uniformly distributed, it seems natural to add ψ_p^2 to the sum of the squared output errors J in a weighted manner. The resulting regularized problem is

$$\begin{aligned} \min_{\theta} J(\theta) + \lambda \cdot \psi_p(\theta)^2 \\ \text{s.t. Dynamics in (2),} \end{aligned} \quad (7)$$

where λ serves as regularization strength. Exemplarily, a resulting state trajectory is illustrated in Fig. 3.

Remark. The regularization technique leads to the stabilization of the local poles due to its relocation

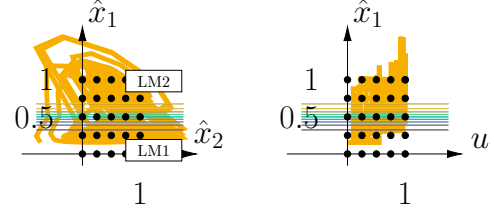


Figure 3: State trajectory after the optimization for $\lambda = 10$ on $n_g=125$ grid points in three-dimensional extended input/state space.

effect on data points in both LMs:

$$\begin{aligned} \max_i |\mu_i(\underline{A}_{LM,1})| &= 0.78 < 1, \\ \max_i |\mu_i(\underline{A}_{LM,2})| &= 0.89 < 1 \end{aligned}$$

Additionally, regularization enables the LMs to explain the dynamical character of the process: oscillations are modeled with local complex eigenvalues, which was not accomplished with the trajectories visible in Fig. 1. This demonstrates that the method increases both robustness *and* interpretation of the model, whereas a tradeoff regarding modeling performance is achieved.

Next, we compare the evolution of loss J and ψ_p for the regularized and unregularized problem during the first split optimization, see Fig. 4.

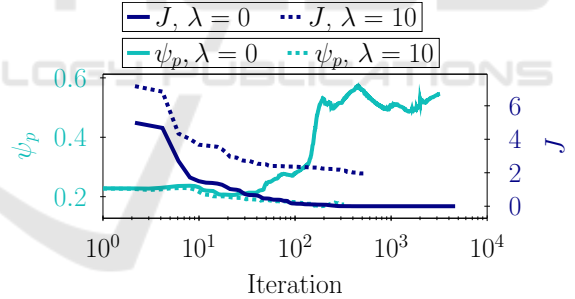


Figure 4: Comparison of regularization effect on ψ_p and J . Both ψ_p evolutions contain the initial space-filling at the first iteration with $\psi_p=0.23$.

As already mentioned, in the unregularized case, ψ_p grows along the iterations, whereas the regularized curve shows a slight decrease. This means that the state trajectory *expands* during optimization. Consequently, due to regularization, the loss cannot be reduced to $J \approx 0$, because of the tradeoff between output error fit and space-filling of the distribution. At termination, it reaches a level of about $J \approx 2$. Furthermore, the early-stopping effect caused by the penalty term is visible. The regularized problem is terminated after 324 iterations whereas the baseline runs till iteration 3232. This enables major savings in training time.

For the choice of the hyperparameter λ , a grid search for values between 10^{-4} and 10^4 is carried out. Figure 5 visualizes J , ψ_p and the number of iterations. The figure shows an increase of J for stronger reg-

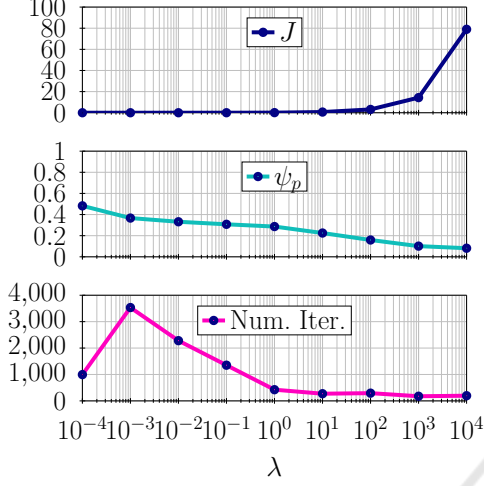


Figure 5: Loss, space-filling and iterations after optimization of problem (7) for various regularization strengths λ .

ularization, whereas ψ_p can only be changed moderately due to the dynamic relations between data points. Since the initial state trajectory (iteration = 0, see light blue curve in Fig. 1) shows suitable space-filling properties, it is sensible to choose λ such that the final state trajectory yields a similar value for ψ_p . As Fig. 4 shows, the initial value for $\psi_p = 0.23$ is obtained for a linear model in balanced realization. This amount of space-filling can be roughly matched at convergence after the first split for the choice of regularization strength of $\lambda = 8$. Independent of the space-filling properties, $\lambda > 1$ are recommended to realize maximum advantage w.r.t. computational demand from the early-stopping (see Fig. 5, bottom diagram).

4.2 Desired Amount of Space-Filling

The proposed strategy requires a hyperparameter study to find a good value for λ . If the desired value for space-filling is known a priori, e.g., keeping ψ_p from the initial model, an alternative approach is to penalize the *deviation* between the current ψ_p and the desired one (named $\psi_{p,Target}$). By this, it represents as a desired degree of regularization. This regularized nonlinear problem can be stated as follows,

$$\min_{\theta} J(\theta) + \lambda \cdot \left(\psi_p(\theta) - \psi_{p,Target} \right)^2 \quad (8)$$

s.t. Dynamics in (2).

Here, a large value is chosen for the regulariza-

tion strength λ , since it simply determines the tightness for matching $\psi_{p,Target}$. For the demo process $\psi_{p,Target}=0.23$ is chosen, as it reflects the space-filling of the initial model. Figure 6 shows the results of the study on λ for problem (8). The loss J and the space-filling quality ψ_p are nearly independent of the choice of λ . Therefore, it can be determined according to the other criteria. Consequently, it is chosen as 10^3 , being a tradeoff between initial space-filling and training time, measured by the number of iterations.

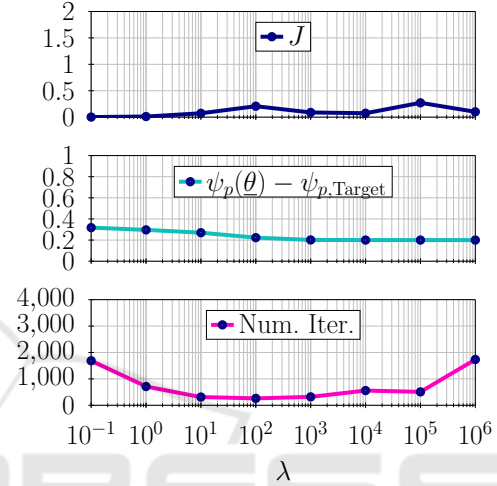


Figure 6: Optimization results of problem (8) for various values of λ .

5 BENCHMARK DATASET

Next, we investigate the proposed regularization strategy on a well-known system identification benchmark, namely the Bouc-Wen Hysteretic System. For details on the benchmark, we refer to Schoukens and Noël (2017).

For brevity, we only show the results for the *Desired Amount of Space-Filling* approach, see problem (8) in Section 4.2. For the study, the space-filling goal is chosen as $\psi_{p,Target}=0.25$. As for the demo process, the regularization strength is set to $\lambda = 10^3$.

Remark. The termination of LOLIMOT is based on the *validation* error. It is terminated when no significant improvements (25 % of a normalized RMSE corresponding to an SNR of 40 dB) compared to the best loss in all previous iterations can be accomplished for three iterations. Then the simplest model within this error range is selected.

In Fig. 7, we compare the regularized and unregularized modeling cases regarding the RMSE of each split on training, validation, and test dataset.

According to the termination criterion, the best

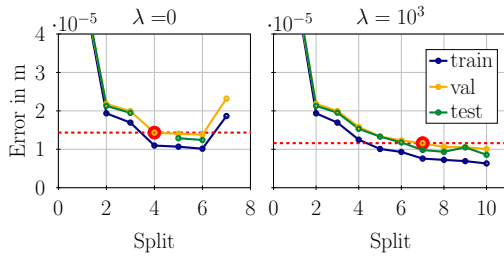


Figure 7: RMSE during LOLIMOT training. The *best model* according to the validation error termination is marked in red.

model for the unregularized case ($\lambda=0$) has four splits because the validation error increases due to the seventh split. This is caused by the splitting procedure of LOLIMOT (see Sec. 2.1). The addition of a LM, $n_{LM,x} \rightarrow n_{LM,x} + 1$, requires updating the centers and standard deviations of the Gaussians. Due to the normalization, the validity functions $\Phi_j^{[x]}$ are modified, which in turn influences the state trajectory and thus the output error. A poor initialization of the consecutive optimization of the split results and cannot be compensated. Consequently, the training error of the seven-split model is higher than the six-split model.

When evaluating the unregularized models on test data, the models obtained by split numbers four and seven show unstable behavior. In those cases, no sensible RMSE can be obtained, which is why their values are omitted from the figure. Within the regularized model ensemble, no instability could be detected regarding test data. Consequently, the robustness of LOLIMOT is improved.

Furthermore, it can be observed that regularization allows LOLIMOT to perform seven splits for the best model, compared to only four splits in the unregularized case. Regularization enables the algorithm to process the training dataset information with more LMs. Evaluation on the benchmark's test dataset yields an RMSE of $0.981 \cdot 10^{-5}$ m for the regularized best model, whereas the unregularized one became unstable. In contrast to the stable splits, it can be stated that space-filling regularization enables better models. In comparison to other approaches, the achieved result is among the best for this benchmark (Nonlinear Benchmark Working Group, 2025).

6 CONCLUSIONS

In this paper, we investigated the effect of nonlinear optimization on the extended input/state space data point distribution of nonlinear state space models. Two regularization approaches for penalizing poor space-filling quality are derived and tested, resulting

in models with more meaningful and accurate local models. This is achieved by more local estimation and less interpolation as well as stable local behavior. Furthermore, higher modeling performance could be accomplished and the number of iterations could be reduced per optimization. Future lines of research will focus on the stable extrapolation behavior due to space-filling enforcement in the local model state space network. Moreover, the applicability of space-filling metrics with higher-dimensional settings will be addressed.

REFERENCES

- Belz, J., Munker, T., Heinz, T. O., Kampmann, G., and Nelles, O. (2017). Automatic modeling with local model networks for benchmark processes. *IFAC-PapersOnLine*, 50(1):470–475. 20th IFAC World Congress.
- Bemporad, A. (2024). An l-bfgs-b approach for linear and nonlinear system identification under ℓ_1 and group-lasso regularization.
- Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.
- Forgione, M. and Piga, D. (2021). Model structures and fitting criteria for system identification with neural networks.
- Garulli, A., Paoletti, S., and Vicino, A. (2012). A survey on switched and piecewise affine system identification. *IFAC Proceedings Volumes*, 45(16):344–355. 16th IFAC Symposium on System Identification.
- Herkersdorf, M. and Nelles, O. (2025). Online and offline space-filling input design for nonlinear system identification: A receding horizon control-based approach. *arXiv preprint arXiv:2504.02653*.
- Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86.
- Liu, Y., Tóth, R., and Schoukens, M. (2024). Physics-guided state-space model augmentation using weighted regularized neural networks.
- Ljung, L. (1999). *System identification (2nd ed.): theory for the user*. Prentice Hall PTR, USA.
- Ljung, L., Andersson, C., Tiels, K., and Schön, T. B. (2020). Deep learning and system identification. *IFAC-PapersOnLine*, 53(2):1175–1181. 21st IFAC World Congress.
- Luenberger, D. (1967). Canonical forms for linear multivariable systems. *Automatic Control, IEEE Transactions on*, AC-12:290 – 293.
- McKelvey, T., Akcay, H., and Ljung, L. (1996). Subspace-based multivariable system identification from frequency response data. *IEEE Transactions on Automatic Control*, 41(7):960–979.
- Nelles, O. (2020). *Nonlinear System Identification From Classical Approaches to Neural Networks*, Fuzzy

- Models, and Gaussian Processes*. Springer International Publishing; Imprint: Springer, 2020. 2nd ed. 2020., Cham.
- Nonlinear Benchmark Working Group (2025). Nonlinear benchmark models for system identification. <https://www.nonlinearbenchmark.org/benchmarks>. Accessed: 2025-05-19.
- Paduart, J., Lauwers, L., Swevers, J., Smolders, K., Schoukens, J., and Pintelon, R. (2010). Identification of nonlinear systems using polynomial nonlinear state space models. *Automatica*, 46:647–656.
- Pillonetto, G., Chen, T., Chiuso, A., De Nicolao, G., and Ljung, L. (2022). *Regularization for Nonlinear System Identification*, pages 313–342. Springer International Publishing, Cham.
- Pintelon, R. and Schoukens, J. (2012). *System Identification: A frequency Domain Approach*. Wiley / IEEE Press, 2 edition. IEEE Press, Wiley.
- Rotondo, D., Puig, V., Nejjari, F., and Witczak, M. (2015). Automated generation and comparison of takagi-sugeno and polytopic quasi-lpv models. *Fuzzy Sets and Systems*, 277.
- Schoukens, M. and Noël, J. (2017). Three benchmarks addressing open challenges in nonlinear system identification. *IFAC-PapersOnLine*, 50(1):446–451. 20th IFAC World Congress.
- Schüssler, M. (2022). *Machine learning with nonlinear state space models*. PhD thesis, Universität Siegen.
- Schüssler, M., Munker, T., and Nelles, O. (2019). Local model networks for the identification of nonlinear state space models. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 6437–6442.
- Scott, D. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. A Wiley-interscience publication. Wiley.
- Smits, V. and Nelles, O. (2024). Space-filling optimized excitation signals for nonlinear system identification of dynamic processes of a diesel engine. *Control Engineering Practice*, 144:105821.
- Suykens, J. A. K., De Moor, B. L. R., and Vandewalle, J. (1995). Nonlinear system identification using neural state space models, applicable to robust control design. *International Journal of Control*, 62(1):129–152.
- Van Overschee, P. and De Moor, B. (1995). A unifying theorem for three subspace system identification algorithms. *Automatica*, 31(12):1853–1864. Trends in System Identification.
- Verriest, E. and Kailath, T. (1983). On generalized balanced realizations. *IEEE Transactions on Automatic Control*, 28(8):833–844.