Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Identification of State-space Linear Parameter-varying Models Using Artificial Neural Networks ⋆

Yajie Bao * Javad Mohammadpour Velni ** Aditya Basina *** Mahdi Shahbakhti ****

* School of Electrical and Computer Engineering, The University of Georgia, Athens, GA 30602, USA (e-mail: yb18054@uga.edu).
** School of Electrical and Computer Engineering, The University of Georgia, Athens, GA 30602, USA (e-mail: javadm@uga.edu).
*** Dept. of Mechanical Engineering, Michigan Tech University, Houghton, MI 49931, USA (e-mail: abasina@mtu.edu)
**** Dept. of Mechanical Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada (e-mail: mahdi@ualberta.ca)

Abstract: This paper presents an integrated structure of artificial neural networks, named state integrated matrix estimation (SIME), for linear parameter-varying (LPV) model identification. The proposed method simultaneously estimates states and explores structural dependency of matrix functions of a representative LPV model only using inputs/outputs data. The case with unknown (unmeasurable) states is circumvented by SIME using two estimators of the same state: one estimator represented by an ANN and the other obtained by LPV model equations. Minimizing the difference between these two estimators, as part of the cost function, is used to guarantee their consistency. The results from a complex nonlinear system, namely a reactivity controlled compression ignition (RCCI) engine, show high accuracy of the state-space LPV models obtained using the proposed SIME while requiring minimal hyperparameters tuning.

Keywords: System identification; Linear parameter-varying systems; Artificial neural networks.

1. INTRODUCTION

Identification of linear parameter-varying (LPV) models in state-space (SS) form has attracted a lot of interest because such models allow capturing nonlinearities and time-varying behavior in a system using a linear structure such that linear controller synthesis techniques can be applied (see Rizvi et al. (2018) and references therein). Identification of LPV-SS models is in general challenging especially when states are unknown. Recent efforts have been devoted mostly to employ supervised machine learning methods to tackle this issue.

Existing studies on data-driven methods for global identification of LPV-SS models can be categorized into direct prediction-error minimization (PEM) methodologies, set membership approaches (SM) and global subspace and realization based techniques (SID) (Cox, 2018). PEM can be further divided into gradient-based and expectation maximization (EM) approaches. The majority of the current LPV identification methods assume affine scheduling dependency with known basis, which restricts the complexity of a representation. Rizvi et al. (2018) used non-parametric kernel methods to learn scheduling dependency, which faces problems such as kernel selection and computational complexity. Moreover, SID constructs SS models from an identified specific IO structure by either a direct realization or a projection to first estimate state sequence and then estimate system matrices. The disadvantage with this strategy is that the error (either from structural uncertainty or algorithmic uncertainty) of the first stage can affect the performance of the second stage and the combined influence on model performance is difficult to analyze for the downstream work such as controller design. Additionally, all current SID methods suffer from the curse of dimensionality.

The objective of this paper is to develop an integrated approach to estimate states and matrix functions simultaneously. Specifically, we propose the use of artificial neural networks (ANNs) to represent the functions in the model. On one hand, the advantages of ANN representing functions are sufficiently argued in (Luzar and Czajkowski, 2016); on the other hand, the development of libraries such as Keras and Tensorflow (Chollet et al., 2015) has tremendously accelerated the implementation of deep neural networks. Compared with affine scheduling dependency with known basis, ANN can model arbitrary structural dependency without specifying basis; compared with kernelized methods, ANN can avoid kernel selection and reduce computational complexity. Although ANN structure design requires domain knowledge, sufficiently expressive ANN can learn a good scheduling dependency from data.

Using ANNs to identify mathematical models of systems is common in model identification problems because of their

⋆ This research was financially supported by the United States National Science Foundation under award #1762595.

high expressiveness and flexibility. Some researchers have used neural networks to represent part of the system that is difficult to describe in an analytical way (Saadat et al. (2004); Lu et al. (2008); Previdi and Lovera (2004)). However, this approach requires sufficient knowledge about the system which may not be available in practice. Other research efforts represented the whole system by neural networks. One typical study is (Luzar and Czajkowski, 2016) that introduced the State-Space Neural Network (SSNN) and developed a toolbox that can transform the SSNN parameters into state-space matrices, but the error introduced by the transformation was not discussed in (Luzar and Czajkowski, 2016). Moreover, Verdult et al. (2002) used neural networks to represent a system whose model is a weighted combination of local linear SS models. Using this special model structure, the problem of determining a sufficient number of local models was not addressed.

In this work, firstly, we propose an end-to-end machine learning method to incorporate state estimation module and matrix function estimation module into one integrated model (named SIME). Specifically, we aim to minimize the difference between two estimates (represented by neural networks) of the same states for state estimation module, minimize the output estimation error for matrix function module, and use the weighted sum of these two objectives as the total loss function of SIME. By optimizing SIME, we can estimate states and matrix functions simultaneously. Secondly, we show how the SIME-based approach can be used to boost accuracy with known states. Research shows that over-parameterization contributes to the outstanding performance of ANN but also causes overfitting problem while the idea of SIME can provide regularization for ANN training.

The application of the proposed method is demonstrated on an advanced combustion engine that exhibits highly nonlinear and complex combustion process. In particular, the SIME will be designed and tested for a reactivity controlled compression ignition (RCCI) engine that represents the frontier of the research for high-efficiency, low-emission internal combustion engines. The control and calibration of RCCI engines for transient operations is challenging and time-consuming (Raut et al., 2018). The development of first principle physical control-oriented models (COMs) (Khodadadi Sadabadi et al., 2016) for RCCI is time consuming, while state-of-the-art COMs can not predict RCCI cyclic variability and engine-out emissions. The development of data-driven LPV-SS models is of high interest for RCCI controls to: (i) reduce COM development time, (ii) simplify controller design versus using complex nonlinear RCCI controllers (Indrajuana et al., 2016), and (iii) obtain robust control performance for broad engine operating conditions instead of using switched gain-scheduled linear controllers (Raut et al., 2018).

In our prior work (Irdmousa et al., 2019), we developed a data-driven COM for predicting RCCI combustion phasing (CA50), using the method of least-square support vector machines (LS-SVM). Two limitations of our prior work reported in (Irdmousa et al., 2019) were the high computational cost for model training/identification and the need for knowing the states of the RCCI engine. To the best of the authors' knowledge, this paper presents the *first study undertaken to create a computationally-efficient data-driven SSNN model for RCCI engines* to predict CA50 and engine load (IMEP) in a new LPV-SS platform for RCCI control design.

The rest of the paper is organized as follows: Section 2 provides LPV-SS model identification problem formulation. The structure and cost function of SIME will be explained in Section 3. Section 4 shows the adaptation of SIME to the scenario with known states. Section 5 will discuss and show the model identification results. Concluding remarks are finally provided in Section 6.

2. PROBLEM FORMULATION

A discrete-time, state-space (SS) innovation-type noise model of an LPV system can be expressed as

$$x_{k+1} = A(p_k)x_k + B(p_k)u_k + K(p_k)e_k, \tag{1}$$
$$y_k = C(p_k)x_k + D(p_k)u_k + e_k, \tag{2}$$

where $p_k \in \mathbb{P} \subset \mathbb{R}^{n_p}$, $u_k \in \mathbb{R}^{n_u}$, $x_k \in \mathbb{R}^n$, $e_k \in \mathbb{R}^{n_y}$, and $y_k \in \mathbb{R}^{n_y}$ denote the scheduling variables, inputs, states, stochastic white noise process, and outputs of the system at time instant k, respectively, and A, B, C, D, and K are smooth matrix functions of p_k. Equivalently, we have

$$x_{k+1} = \tilde{A}(p_k)x_k + \tilde{B}(p_k)u_k + K(p_k)y_k, \tag{3}$$
$$y_k = C(p_k)x_k + D(p_k)u_k + e_k, \tag{4}$$

where $A(p_k) = \tilde{A}(p_k) + K(p_k)C(p_k)$ and $B(p_k) = \tilde{B}(p_k) + K(p_k)D(p_k)$. The definition of structural observability for this LPV-SS representation is given by Rizvi et al. (2018) Definition 3.1. The LPV-SS model identification problem is to estimate states (if they are unknown), $\tilde{A}(p_k)$, $\tilde{B}(p_k)$, $C(p_k)$, $D(p_k)$ and $K(p_k)$ given the measurements $\mathcal{D} = \{u_k, y_k, p_k\}_{k=1}^N$.

State Estimation

In (Rizvi et al., 2018), authors show that x_k can be expressed as

$$x_k = \underbrace{\left(\prod_{i=1}^{d} \tilde{A}(p_{k-i})\right) x_{k-d}}_{\mathcal{X}_p^d(k)} + \mathcal{R}_p^d(k)\overline{u}_k^d + \mathcal{V}_p^d(k)\overline{y}_k^d, \tag{5}$$

where $\mathcal{R}_p^d(k)$ is a d-step backward reachability matrix at time k along the scheduling trajectory p, $\overline{u}_k^d := \left[u_{k-d}^{\mathrm{T}} \cdots u_{k-1}^{\mathrm{T}}\right]^{\mathrm{T}}$ denotes the past inputs and $\overline{y}_k^d$ is defined similarly. Ideally, by choosing $\mathcal{X}_p^d(k) \approx 0$, we have $x_k = f(\overline{p}_k^d, \overline{u}_k^d, \overline{y}_k^d)$. Additionally, the equality holds when $d = k$. Assuming x_0 is known, it is impractical to use the whole past trajectory. Instead, we use a truncated trajectory with a length of d. Equation (5) shows that x_k can be expressed as a function of past states, inputs and outputs, which provides a basis for "state integrated matrix estimation" (SIME) and will be discussed in the next section.

3. STATE INTEGRATED MATRIX FUNCTIONS ESTIMATION

In this section, we introduce the structure and cost functions of SIME in detail. The complete computing graph of

SIME is shown in Figure 1 and divided into two sub-graphs described next.

3.1 Computation Graph of State Estimation

Inspired by (5), we use a neural network (NN_F in Figure 1) to directly represent the relationship between x_k and $\overline{z}_k^d := \left[\overline{p}_k^{d\mathrm{T}} \ \overline{u}_k^{d\mathrm{T}} \ \overline{y}_k^{d\mathrm{T}} \ p_k^{\mathrm{T}} \ u_k^{\mathrm{T}} \ \right]^{\mathrm{T}}$ (note that $\overline{z}_k^d$ represents the past and current information at time instant k). By NN_F, we have an estimator of states, i.e., Estimator 2 in Figure 1. Using Estimator 2, we can obtain $\hat{x}_k^{(2)}$; then, using (3) and by representing $\tilde{A}(p_k)$, $\tilde{B}(p_k)$, and $K(p_k)$ with 3 different neural networks, we can obtain $\hat{x}_{k+1}^{(1)}$. Furthermore, we can directly use NN_F to obtain another estimation ($\hat{x}_{k+1}^{(2)}$) for x_{k+1}. We note that $\hat{x}_{k+1}^{(1)}$ and $\hat{x}_{k+1}^{(2)}$ should be consistent as they estimate the same states. Based on this idea, we use the Mean Square Error (MSE) between these two estimates as a regularization term for SIME.

Interpretation of the state estimation in SIME: According to the universal approximation theorem (see Csáji (2001)), Estimator 2 can capture all the nonlinearities of the system. Specially, Estimator 2 are constructed such that (5) can be well represented. However, Estimator 1 adds a constraint that the model needs to be in the form of (3). By minimizing

$$\mathbb{E}[x_k^{(2)}(\overline{p}_k^d, \overline{u}_k^d, \overline{y}_k^d, p_k, u_k) - x_k^{(1)}(p_k, u_k, y_k, x_{k-1}^{(2)})]^2, \quad (6)$$

Estimator 1 approximates the "true states" (estimation from Estimator 2) with high accuracy while ensuring the LPV representation of the model. Considering the relationship between mutual information and MSE shown in (Guo et al., 2005), Estimator 2 aims to preserve as much historical information that is useful to determine the states as possible from the perspective of information theory. Moreover, the number of states are determined by data using cross-validation in (Kohavi et al., 1995), a technique in machine learning utilized to tune hyperparameters, which is not related to an identified LPV-IO model. In this way, the curse of dimensionality faced by SID can be moderated.

3.2 Computation Graph of Output Estimation

After obtaining the estimation of states, we can use neural networks to represent $C(p_k)$ and $D(p_k)$ and estimate y_k from (4) using $\hat{x}_k^{(2)}$. Similarly, we can determine $\hat{y}_{k+1}$ using $\hat{x}_k$. In particular, we use $\hat{x}_k^{(1)}$ instead of $\hat{x}_k^{(2)}$ to train $NN_{\tilde{A}}$, $NN_{\tilde{B}}$ and NN_K. Then, we use both MSE between y_{k+1} and $\hat{y}_{k+1}$ and MSE between y_k and $\hat{y}_k$ to train NN_C and NN_D. As shown by Lemma 3.1 in Rizvi et al. (2018), there exists a function f such that for any scheduling trajectories $y_k = f(u_k, p_k, e_k, \overline{z}_k^d, \overline{e}_k^d)$. Correspondingly, we can claim there exists a parameterization of SIME that can represent this f.

Summary of the SIME model: Computation graphs of state and output estimation parameterize the estimates of states and outputs using ANN. ANN can not only approximate any family of functions but also can incorporate prior knowledge of systems when designing the structure of neural networks. Moreover, the cost function of SIME is defined as the difference between estimations and true values and consists of three parts: MSE of $\hat{x}_{k+1}^{(1)}$ and $\hat{x}_{k+1}^{(2)}$ (L_1), MSE of $\hat{y}_k$ and y_k (L_2) and MSE of $\hat{y}_{k+1}$ and y_{k+1} (L_3). Additionally, the weights of the three losses can be tuned using cross-validation.

3.3 Computation Graph of Output Prediction

After training the integrated model, we can predict the output to validate our model. First, given $\overline{z}_{l-1}^d$ and p_{l-1}, we can estimate x_{l-1} by NN_F, $\tilde{A}(p_{l-1})$ by $NN_{\tilde{A}}$, $\tilde{B}(p_{l-1})$ by $NN_{\tilde{B}}$ and $K(p_{l-1})$ by NN_K. Then, using the recurrent equation of states, we obtain the estimation of $x_l^{(1)}$. Next, given p_l, we can estimate $C(p_l)$ by NN_C, and $D(p_l)$ by NN_D. Finally, using (4), we can determine the prediction of y_l. It is noted that we can readily extract the matrix functions by running the corresponding neural networks with $\{p_k\}_{k=1}^N$ as the inputs.

4. LPV MODEL ESTIMATION WITH KNOWN STATES

Given the states, the LPV model identification problem is reduced to a "regression problem" (Bamieh and Giarre (2002)), as the vector x_k in $\mathcal{D} = \{u_k, x_k, y_k, p_k\}_{k=1}^N$ is now known. Using Estimator 1 to estimate $\tilde{A}(p_k)$, $\tilde{B}(p_k)$ and $K(p_k)$ becomes feasible. However, since the true A, B and K are inaccessible and we only have data $\mathcal{D}$ for function approximation, overparameterization [1] is commonly used in ANN to achieve small prediction error on training set while regularization terms are added to constrain the solution space and improve the generalization ability of the model. The loss term L_1 in SIME can serve as a regularization term, which is validated using experiments on real engine data in Section 5.2.

Specifically, with known states, we use x_k instead of $\hat{x}_k^{(2)}$ to construct $\hat{x}_{k+1}^{(1)}$ from (3) and use NN_F to construct $\hat{x}_k^{(2)}$ and $\hat{x}_{k+1}^{(2)}$. The cost function similarly consists of three parts: MSE of $\hat{x}_{k+1}^{(1)}$ and $\hat{x}_{k+1}^{(2)}$ (L_1), MSE of $\hat{x}_k$ and x_k (L_2) and MSE of $\hat{x}_{k+1}$ and x_{k+1} (L_3).

5. APPLICATION OF THE PROPOSED LPV-SS IDENTIFICATION METHOD TO RCCI ENGINES

In this section, we demonstrate the accuracy of SIME-based LPV-SS models for reactivity controlled compression ignition (RCCI) engines.

The data is collected from a high fidelity simulation model (Raut et al., 2018) that was experimentally validated with the data from a 2-liter, 4-cylinder GM engine. It is assumed that the states of the model are not available for measurements. The control inputs, scheduling variable, and measurement outputs are as follows:

$$U = [SOI \quad FQ]^{\mathrm{T}},$$
$$p = [PR]^{\mathrm{T}},$$
$$Y = [CA50 \quad IMEP]^{\mathrm{T}},$$

[1] Theoretical analysis of overparameterized neural networks can be found in (Li and Liang, 2018).

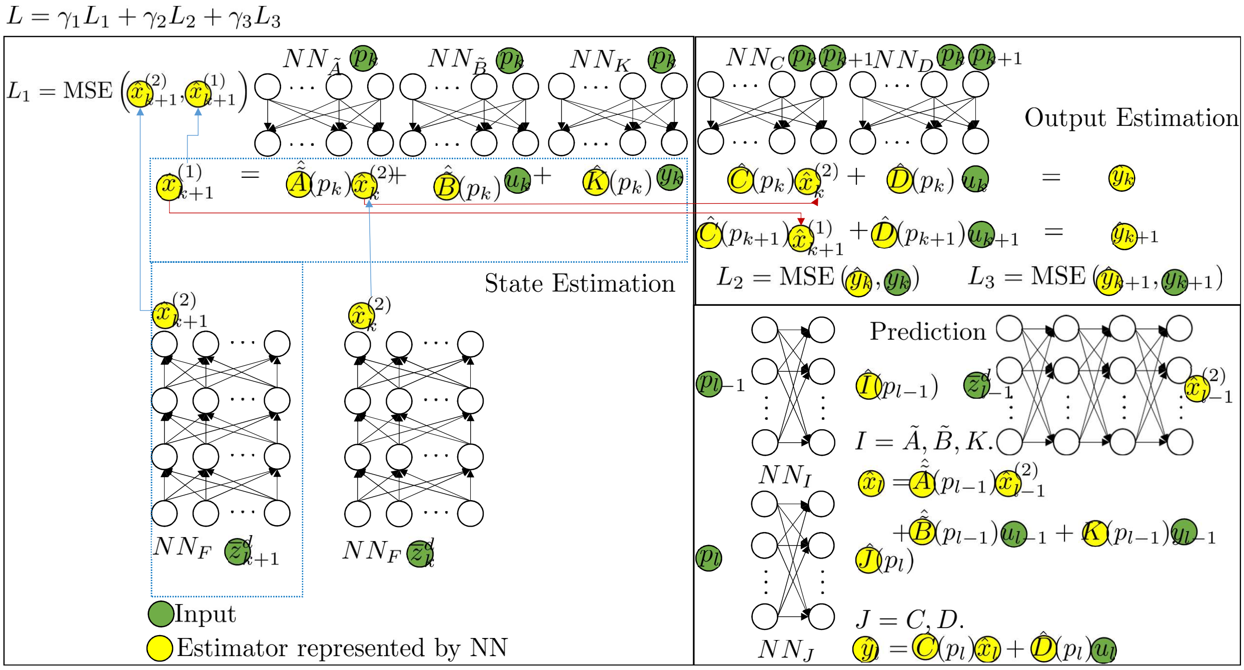
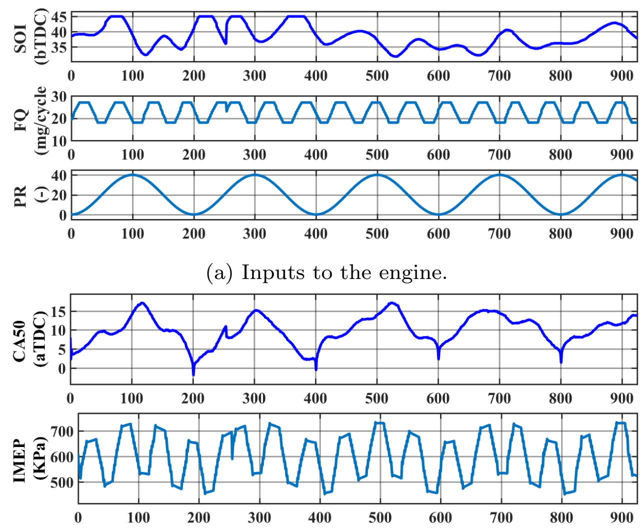


Fig. 1. The complete computing graph of SIME. NN_I ($I = \tilde{A}, \tilde{B}, K, C, D, F$) are used to distinguish neural networks. The green and yellow circles represent the inputs and outputs of their adjacent neural networks respectively (except for $x^{(2)}_k$ and $x^{(2)}_{k+1}$ which are computed by NN_F). The red lines show the connections between computing graphs of state and output estimation modules.

making the system to be identified a two-input/two-output (TITO) system. The control inputs include fuel quantity (FQ) and start of injection (SOI) for injecting the high reactive fuel (n-heptane) into the cylinder. The RCCI combustion phasing for the crank angle of 50% fuel burnt ($CA50$) is controlled by adjusting SOI, while the engine indicated mean effective pressure ($IMEP$) is controlled by adjusting FQ. RCCI is a dual fuel engine in which the ratio of two fuels energy is characterized by premixed ratio (PR) (Raut et al., 2018). The engine operation is highly dependent on PR and exhibits a highly nonlinear behavior as PR changes. Thus, PR is used as the scheduling variable in the LPV model of the RCCI engine.

5.1 LPV-SS Model Identification with Unknown States

We generated an input/output data set using the signals shown in Figure 2. The data set contains 926 operating points and is split into a training set and a testing set with a splitting ratio of 65%/35%.



(a) Inputs to the engine.

(b) Outputs from the engine.

Fig. 2. Input/output data set used for LPV-SS model learning of RCCI engine.

Technical Details of ANN: We use a 4-layer (including the input and output layers) fully-connected neural network to represent each of $\tilde{A}(p_k)$, $\tilde{B}(p_k)$, $C(p_k)$, and $K(p_k)$, resulting in 4 different neural networks while a 5-layer fully-connected neural network (NN_C in Figure 1) is implemented for $C(p_k)$ to be more expressive. All these networks share the same input p while their outputs depend on the matrix function to be estimated. To obtain the correct shape, we reshape the output layers into the corresponding matrix shape. Additionally, all the hidden layers have 5 units and use Exponential Linear Units [2] (ELU) first introduced in (Clevert et al., 2015) as the activation function. The parameter α of ELU is set to 1.0. However, no activation function was used for the output layer of each neural network, as the range of variables varies significantly.

[2] $f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases}$

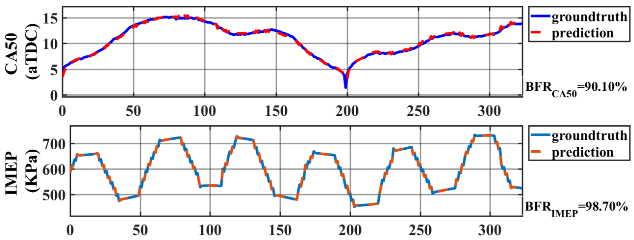


Fig. 3. The output response of the estimated LPV-SS model and the response of the original system on the validation data set.

For Estimator 2 involved in the state estimation, we use a 6-layer (including the input and output layers) fully-connected neural network, as the represented function should be more complex than any matrix function. The inputs and loss function of SIME are summarized as follows: for estimation, the inputs include $\{p_k, u_k, \overline{z}_k, y_k\}_{k=d}^{N-1}$ [3], and the loss function is the weighted sum of MSE of $\hat{x}$ and MSE of $\hat{y}$ (for both time k and $k+1$); for prediction of y_l, the inputs contain $\{p_l, u_l, \overline{z}_l^d\}$. In this experiment, we set the time window size to $d = 2$.

For model optimization, we use Adam optimizer in Keras (Chollet et al., 2015) [4]. The learning rate of Adam is set to be 0.001 and decay to be $1e-6$. All the other parameters of Adam are set as default. We trained SIME for 2,000 epochs with batch size of 1. In particular, larger batch sizes can lower performance if we do not restrict that the samples in one batch have identical scheduling variables, as the neural networks will give one set of parameters for the whole batch. Using cross-validation, the weights of three loss functions were determined to be all 1. The accuracy of identified model can be observed in Figure 3 and the estimation of matrix functions in Figure 4.

The Best Fit Ratio (BFR) used in Figure 3 is calculated according to

$$\mathrm{BFR}(\theta) = 100\% \cdot \max_k \left(1 - \frac{\|y_k - \hat{y}_k(\theta)\|_2}{\|y_k - \bar{y}\|_2}, 0\right), \qquad (7)$$

where $\bar{y}$ denotes the mean value.

5.2 LPV-SS Model Identification with Known States

In this section, we show results of model identification on real engine data with full states measurements. See (Irdmousa et al., 2019) for the description of this experiment setup. The states, inputs, and outputs of the RCCI engine system are

$$X = [CA50 \;\; T_{SOC} \;\; P_{SOC} \;\; IMEP]^{\mathrm{T}},$$
$$U = [PR \;\; SOI \;\; FQ]^{\mathrm{T}},$$
$$Y = [CA50]^{\mathrm{T}}.$$

We also used $PR = 40$ and FQ to be the scheduling variable.

ANN Adjustment: We used *tanh* as activation function for all the ANN layers except for the reshape layers and also used Adam as the optimizer. Additionally, the learning rate of Adam was set to be 0.0001 and decay to be $1e-6$. All the other parameters of Adam were left with the default settings. The adjusted SIME for known states was trained for 1,000 epochs with the batch size of 1.

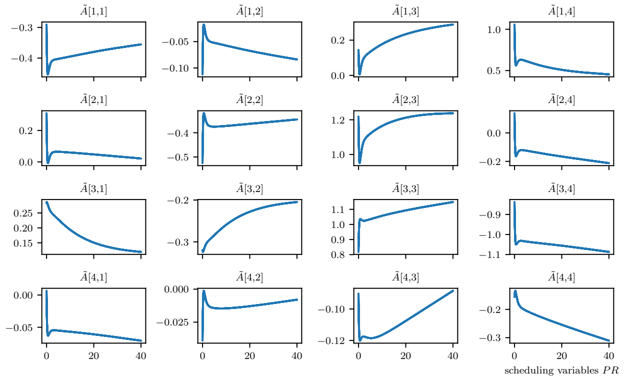


Fig. 4. The estimation of matrix functions as a function of p in the learned LPV-SS model (Take $\tilde{A}$ as an example). All the horizontal axes represent the scheduling variable PR. By approximating a smooth function for each matrix entry, the trained SIME model can predict the states and outputs of a system with high accuracy.

To demonstrate the contribution of the regularization that estimates of Estimator 1 and 2 should be consistent, we conducted a set of comparative experiments. First, we trained the computation graph of state estimation based on Estimator 1 without regularization term and the results are shown in Figure 5(a). Then, we experimented with the regularization term. The cost function of the adjusted SIME is the weighted sum of the MSE of $\hat{x}_{k+1}^{(1)}$ and x_{k+1}, the MSE of $\hat{x}_k^{(2)}$ and x_k, and the MSE of $\hat{x}_{k+1}^{(1)}$ and $\hat{x}_{k+1}^{(2)}$. The weights of the three MSEs are used to reflect their relative importance; those weights were determined to be 1, 0.1, and 0.01, respectively, by cross-validation. Figure 5(b) shows the results of identification with the regularization term and Figure 5(c) reproduced the results given in (Irdmousa et al., 2019) using a least-squares support vector machine (LS-SVM) approach.

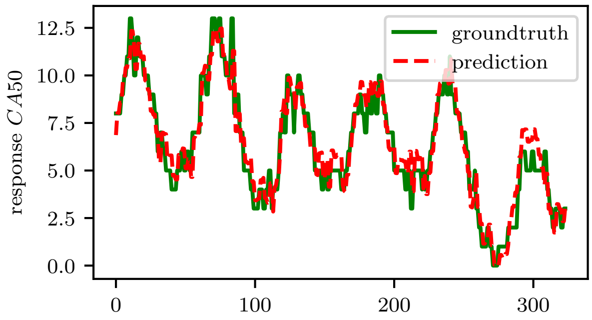
6. CONCLUDING REMARKS

In this paper, an integrated ANN approach (SIME) was proposed to identify an LPV-SS model of systems with unknown states. Specifically, two estimators (to predict states) were constructed: one estimator to take advantage of past system trajectory and another to facilitate matrix functions estimation. The consistency between two estimations of the same states was used to obtain a reasonable state estimation. By minimizing the consistency violation and prediction errors of outputs, SIME was shown to estimate states and matrix functions simultaneously given only input/output data of a system. Moreover, our SIME-based approach was extended to the case with known states by using consistency violation as a regularization term. An experimentally validated high-fidelity RCCI engine model was used to generate data that was then employed to validate the proposed approach. Our results confirmed that SIME can identify LPV-SS models with a high accuracy and very little hyperparameters tuning.

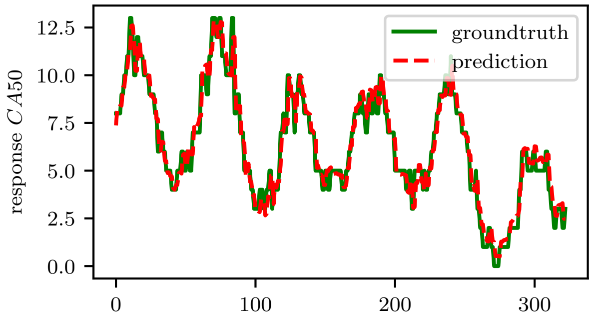
---

[3] We discard a few data points as a result of $\overline{z}_k^d$ and $\hat{y}_{k+1}$.

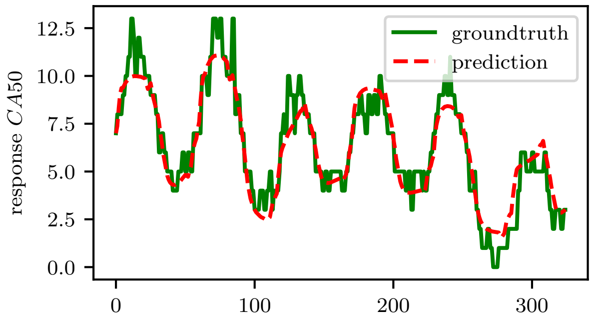
[4] All the codes are implemented in Keras (Chollet et al., 2015).



(a) BFR$_{CA50}$ = 66.73%



(b) BFR$_{CA50}$ = 72.91%



(c) BFR$_{CA50}$ = 64.71%

Fig. 5. Output estimation using identified LPV-SS model on real RCCI engine data with full state measurements. Solid and dashed lines show respectively the measurement and model prediction. Subplot (a) shows the results of regression using ANN without regularization; subplot (b) shows the results of LPV-SS regression using ANN with regularization; subplot (c) shows the result of our previous work (Irdmousa et al., 2019).

REFERENCES

Bamieh, B. and Giarre, L. (2002). Identification of linear parameter varying models. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 12(9), 841–853.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Clevert, D.A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Cox, P.B. (2018). *Towards efficient identification of linear parameter-varying state-space models*. Ph.D. thesis, Ph. D. dissertation, Eindhoven University of Technology.

Csáji, B.C. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24, 48.

Guo, D., Shamai, S., and Verdú, S. (2005). Mutual information and minimum mean-square error in gaussian channels. *IEEE transactions on information theory*, 51(4), 1261–1282.

Indrajuana, A., Bekdemir, C., Luo, X., and Willems, F. (2016). Robust multivariable feedback control of natural gas-diesel rcci combustion. *IFAC-PapersOnLine*, 49(11), 217–222.

Irdmousa, B.K., Rizvi, S.Z., Mohammadpour Velni, J., Naber, J.D., and Shahbakhti, M. (2019). Data-driven modeling and predictive control of combustion phasing for rcci engines. In *2019 American Control Conference (ACC)*, 1617–1622.

Khodadadi Sadabadi, K., Shahbakhti, M., Bharath, A.N., and Reitz, R.D. (2016). Modeling of combustion phasing of a reactivity-controlled compression ignition engine for control applications. *International Journal of Engine Research*, 17(4), 421–435.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, 1137–1145. Montreal, Canada.

Li, Y. and Liang, Y. (2018). Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, 8157–8166.

Lu, B., Choi, H., Buckner, G.D., and Tammi, K. (2008). Linear parameter-varying techniques for control of a magnetic bearing system. *Control Engineering Practice*, 16(10), 1161–1172.

Luzar, M. and Czajkowski, A. (2016). LPV system modeling with SSNN toolbox. In *2016 American Control Conference (ACC)*, 3952–3957. doi: 10.1109/ACC.2016.7525530.

Previdi, F. and Lovera, M. (2004). Identification of non-linear parametrically varying models using separable least squares. *International Journal of Control*, 77(16), 1382–1392.

Raut, A., Bidarvatan, M., Borhan, H., and Shahbakhti, M. (2018). Model predictive control of an rcci engine. In *2018 Annual American Control Conference (ACC)*, 1604–1609. doi:10.23919/ACC.2018.8431172.

Raut, A., Irdmousa, B., and Shahbakhti, M. (2018). Dynamic modeling and model predictive control of an rcci engine. *Control Engineering Practice*, 81, 129 – 144. doi:https://doi.org/10.1016/j.conengprac.2018.09.004. URL `http://www.sciencedirect.com/science/article/pii/S0967066118305082`.

Rizvi, S.Z., Mohammadpour Velni, J., Abbasi, F., Tóth, R., and Meskin, N. (2018). State-space LPV model identification using kernelized machine learning. *Automatica*, 88, 38–47.

Saadat, S., Buckner, G.D., Furukawa, T., and Noori, M.N. (2004). An intelligent parameter varying (IPV) approach for non-linear system identification of base excited structures. *International Journal of Non-Linear Mechanics*, 39(6), 993–1004.

Verdult, V., Ljung, L., and Verhaegen, M. (2002). Identification of composite local linear state-space models using a projected gradient search. *International Journal of Control*, 75(16-17), 1385–1398.