

AN ONLINE TRANSFER LEARNING APPROACH FOR IDENTIFICATION AND PREDICTIVE CONTROL DESIGN WITH APPLICATION TO RCCI ENGINES

Yajie Bao, Javad Mohammadpour Velni*

School of Electrical and Computer Engineering
University of Georgia
Athens, Georgia 30602
Email: {yb18054, javadm}@uga.edu

Mahdi Shahbakhti

Department of Mechanical Engineering
University of Alberta
Edmonton, AB, T6G 1H9, Canada
Email: mahdi@ualberta.ca

ABSTRACT

This paper presents a framework to refine identified artificial neural networks (ANN) based state-space linear parameter-varying (LPV-SS) models with closed-loop data using online transfer learning. An LPV-SS model is assumed to be first identified offline using inputs/outputs data and a model predictive controller (MPC) designed based on this model. Using collected closed-loop batch data, the model is further refined using online transfer learning and thus the control performance is improved. Specifically, fine-tuning, a transfer learning technique, is employed to improve the model. Furthermore, the scenario where the offline identified model and the online controlled system are “similar but not identical” is discussed. The proposed method is verified by testing on an experimentally validated high-fidelity reactivity controlled compression ignition (RCCI) engine model. The verification results show that the new online transfer learning technique combined with an adaptive MPC law improves the engine control performance to track requested engine loads and desired combustion phasing with minimum errors.

1 INTRODUCTION

First principles-based control that offers several advantages for the regulation of nonlinear systems requires an accurate model, while data-driven methods provide efficiency and flexibility on modeling (model learning) highly nonlinear and even

stochastic dynamic systems [1]. The accuracy of the learned models is not only determined by the data-driven methods, but also largely depends on the discrepancy between the application environment and the data-collecting environment, as the generalization of machine learning algorithms is based on the assumption that training and future data share the same feature space and distribution [2]. Collecting data from all possible environments is prohibitive and building a model for each environment from scratch is inefficient, as the process is time consuming and cost intensive. This challenge motivates the development of *transfer learning* to facilitate the model identification of a new environment.

The idea of *transfer learning* that seeks to apply knowledge learned from previous tasks (a.k.a. source tasks denoted as $\mathcal{T}_S$) to new tasks (a.k.a. target tasks $\mathcal{T}_T$) has been extensively studied in the sub-fields of machine learning (e.g., deep learning [3] and reinforcement learning [4]) and employed for tasks such as computer vision [5] and natural language processing [6]. The knowledge can be represented by reusable instances, feature representations, parameters and relational knowledge [7]. A comprehensive survey can be found in [8]. Transferability, i.e., to what extent a source task can help in learning a target task depends on the similarity of the source and target tasks [9]. The similarity can be inferred based on the domain knowledge or from the data of $\mathcal{T}_S$ and $\mathcal{T}_T$, which helps decide on the transfer learning approach.

Among various types of models that are used to repre-

*Address all correspondence to this author.

sent nonlinear dynamic systems, linear parameter-varying, state-space (LPV-SS) models are of great interest for control design purposes. Such models use a linear structure to capture nonlinear and time-varying behaviors of complex systems [10]. Data-driven methods have been increasingly developed for global identification of LPV-SS models using inputs/outputs data. Direct prediction-error minimization (PEM) methods and global subspace and realization-based techniques (SID) are two typical approaches [11]. Both approaches assume affine scheduling dependency with known basis functions, which restricts the complexity of a representation. The authors in [12] used a kernelized least-squares support vector machine (LS-SVM) to capture the dependency structure, which suffers from the kernel selection and computational complexity. Instead, the authors in a very recent work [13] have used artificial neural networks (ANNs) to simultaneously estimate states and explore LPV model structural dependency. Besides universal approximation theorem, neural tangent kernel was introduced in [14] to probe the behavior of ANNs in the so-called large-width limit. Expressive ANNs can learn basis functions from data and provide parametric model estimation. Moreover, deep transfer learning techniques can be adapted for system identification in a new environment.

Deep transfer learning has been well studied especially in computer vision (see [3] for a detailed survey). Fine-tuning is a commonly used technique among the existing methods. Instead of designing and training ANNs from scratch, fine-tuning uses the trained ANN model $\mathcal{M}_S$ for a related $\mathcal{T}_S$ for reference, fixes a part of $\mathcal{M}_S$ that is reusable for the desired model $\mathcal{M}_T$ of $\mathcal{T}_T$ and fine-tunes the specific part of $\mathcal{M}_T$ that is different from $\mathcal{M}_S$. In [15], the authors empirically studied the transition of features from being general to being specific and showed that fine-tuning can produce a boost to generalization.

In this paper, we propose a new online transfer learning in the context of ANN-based LPV-SS model identification using closed-loop data and fine-tuning. Firstly, we will characterize the similarity between $\mathcal{T}_S$ and $\mathcal{T}_T$ based on domain knowledge and introduce the maximum mean discrepancy (MMD) that infers the similarity from data. Secondly, we will discuss the fine-tuning strategies for different degrees of similarity. Moreover, we will introduce our proposed framework to refine models online.

Our proposed online transfer learning method will be assessed on a complex nonlinear thermo-kinetic combustion system known as a reactivity controlled compression ignition (RCCI) engine. RCCI combustion depends on several factors including but not limited to local reactivity gradients inside the combustion chamber, dual fuel ratio, fuel injection timing, fuel injection pressure, number of injections, total injected fuel, intake air temperature and pressure, intake and exhaust valve timings, exhaust gas recirculation (EGR) level and rate, engine speed, and load [16, 17]. Developing broadly applicable physics-based control-oriented RCCI engine models [18–21] is challenging and very time consuming. On the other hand, developing

data-driven RCCI models [13, 22, 23] with many engine variables is also very challenging for broad engine operating conditions. This paper provides novel contributions that will allow adaptive model identification by online transfer learning to improve accuracy of original data-driven RCCI control model for engine control purposes. The results will be demonstrated for RCCI cycle-by-cycle combustion phasing and load control.

The remainder of this paper is organized as follows: Section 2 gives the problem statement and introduces ANN-based LPV-SS identification and adaptive model predictive control (MPC). Similarity characterization and fine-tuning strategies will be discussed in Section 3. Section 4 presents the experiments on two “similar but not identical” RCCI engine models to evaluate the performance of the proposed method. Concluding remarks are finally provided in Section 5.

2 Problem Statement

In machine learning, *domain* $\mathcal{D} = \{\mathcal{X}, P_X(x)\}^1$ is used to describe the input space $\mathcal{X}$ and the associated distribution P_X on the data set; *task* $\mathcal{T} = \{\mathcal{Y}, h\}$ consists of the output space $\mathcal{Y}$ and the mathematical model $h: \mathcal{X} \rightarrow \mathcal{Y}$ to approximate an oracle that knows the correct answers to all questions. Model h can be a deterministic function $f: y = f(x)$ or a distribution $P_{XY}(x, y)$. In this paper, we use *environment* to refer to the oracle. Environment changes when one system changes operating conditions or switches to another system.

Assuming that we have an identified ANN-based LPV-SS model $\mathcal{M}_1$ of a dynamic system $\mathcal{P}_1$ using open-loop data $\mathcal{D}_1$ and a designed controller $\mathcal{C}_1$ based on $\mathcal{M}_1$, the main goal of this paper is to refine the model $\mathcal{M}_1$ online to obtain a good model for a similar but not identical system $\mathcal{P}_2$ using the closed-loop data such that the control performance of $\mathcal{C}_1$ on $\mathcal{P}_2$ can be enhanced. We assume that the new system $\mathcal{P}_2$ is similar to the previous system $\mathcal{P}_1$ such that we can use the existing controller to collect closed-loop data $\mathcal{D}_2$. However, as there are differences between two systems, the distributions of the closed-loop data and previous open-loop data $\mathcal{D}_1$ can be different, which can lead to a deterioration in the performance of the previous model $\mathcal{M}_1$ on the data $\mathcal{D}_2$ of the new system $\mathcal{P}_2$. This problem is known as *transductive learning* or *domain adaptation*, where the domain of source task $\mathcal{D}_S$ and the domain of target task $\mathcal{D}_T$ are different but related while $\mathcal{Y}_S = \mathcal{Y}_T$.

2.1 ANN-based LPV-SS Model Identification

The following discrete-time LPV-SS model with innovation noise is used to describe systems of interest

$$x_{k+1} = A(p_k)x_k + B(p_k)u_k + K(p_k)e_k, \quad (1)$$

$$y_k = C(p_k)x_k + D(p_k)u_k + e_k, \quad (2)$$

¹We use $\mathcal{X}$, X , x and $P_X(x)$ to respectively denote the space, the variable, the sample and the distribution, and hence $x \in X \subseteq \mathcal{X}$.

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

$$x_{k+1} = \tilde{A}(p_k)x_k + \tilde{B}(p_k)u_k + K(p_k)y_k, \quad (3)$$

$$y_k = C(p_k)x_k + D(p_k)u_k + e_k, \quad (4)$$

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 problem of LPV-SS model identification is to estimate states, as well as the matrices $\tilde{A}(p_k)$, $\tilde{B}(p_k)$, $C(p_k)$, $D(p_k)$ and $K(p_k)$ given the measurements $\mathcal{D} = \{u_k, y_k, p_k\}_{k=1}^{N_D}$.

For ANN-based LPV-SS model identification introduced in [13], each of the matrix functions and state vector was represented by a fully-connected ANN. By minimizing the prediction error and the consistency violation between one state estimator represented by an ANN and the other state estimator from (3), State Integrated Matrix Function Estimation (SIME) shown in Figure 1 can provide an accurate LPV-SS model with moderate hyper-parameter tuning.

2.2 Adaptive MPC Design Based on Identified LPV-SS Model

Model predictive control (MPC) design problem for reference tracking is to solve the following multi-constraint optimization problem at each control interval k

$$\min_{\Delta u(k+i|k)} \sum_{i=1}^P \left(\|r(k+i|k) - y(k+i|k)\|_Q^2 + \|\Delta u(k+i|k)\|_R^2 \right) \quad (5)$$

$$\text{s.t. } \Delta u_{\min} \leq \Delta u(k+i|k) \leq \Delta u_{\max}, \quad i = 1, 2, \dots, P \quad (6)$$

$$u_{\min} \leq u(k+i|k) \leq u_{\max}, \quad i = 1, 2, \dots, P \quad (7)$$

$$y_{\min} \leq \hat{y}(k+i|k) \leq y_{\max}, \quad i = 1, 2, \dots, P \quad (8)$$

and the model dynamics in (3)-(4), where P is the (length of) prediction horizon, $r(k+i|k)$ and $y(k+i|k)$ respectively denote reference value and predicted value at i -th prediction horizon step, $\Delta u(k+i|k) = u(k+i|k) - u(k+i-1|k)$ is the incremental input, and $Q \succeq 0$, $R \succ 0$ are tunable weight matrices. Additionally, in this paper, we use $p(k+i|k) = p_k$, $i = 1, 2, \dots, P$. Experiments show that in this way, finite receding horizon MPC can still achieve good control performance with accurately identified model and slowly varying scheduling variables, although the model for predictions at interval j depends on p_j while the future values $p(k+i|k)$ in the prediction horizon are usually unavailable. Stability and feasibility of such LPV-MPC are beyond the scope of this study. Furthermore, Kalman filter is combined with MPC to moderate plant-model mismatch.

3 Similarity and Transfer Learning

In this section, we will introduce the *maximum mean discrepancy* (MMD), a metric to measure the similarity between two environments and *fine-tuning*, a technique for transfer learning with ANN models. Then, we will present the proposed framework of online transfer learning.

3.1 Maximum Mean Discrepancy

The similarity between two environments determines *transferability*, i.e., shared knowledge, and thus affects the transfer learning strategy. For example, a system $\mathcal{P}_1$ is more similar to the second generation $\mathcal{P}_2$ of $\mathcal{P}_1$ than to a system $\mathcal{P}_3$ with a brand-new design. Therefore, an identified model of $\mathcal{P}_1$ can be adapted to $\mathcal{P}_2$ more easily than to $\mathcal{P}_3$. Similarity can be characterized based on the domain knowledge such as the common modules of $\mathcal{P}_1$ and $\mathcal{P}_2$ and the relative complexity of $\mathcal{P}_1$ compared with $\mathcal{P}_3$. This qualitative approach cannot provide transfer bounds that can be used as a stopping condition to avoid negative transfer. To quantify similarity, one approach is to integrate similarity estimation into transfer learning algorithms and another approach is to estimate similarity from data before transfer [9]. We focus on the latter approach to help determine whether or not to transfer.

Estimating similarity of two environments from data can be formulated as a binary hypothesis testing problem. The maximum mean discrepancy (MMD) proposed in [24] for a kernel two-sample test provides a statistic to compare distributions. Given samples $X : \{x_i\}_{i=1}^m \sim p$ and $Y : \{y_j\}_{j=1}^n \sim q$, the empirical estimate of MMD is given by

$$\begin{aligned} \widehat{\text{MMD}}(X, Y) &= \left\| \frac{1}{m} \sum_{i=1}^m \phi(x_i) - \frac{1}{n} \sum_{j=1}^n \phi(y_j) \right\|_{\mathcal{H}}^2 \\ &= \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m k(x_i, x_j) + \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n k(y_i, y_j) - \\ &\quad \frac{2}{mn} \sum_{i=1}^m \sum_{j=1}^n k(x_i, y_j) \end{aligned} \quad (9)$$

where $\mathcal{H}$ denotes a reproducing kernel Hilbert space (RKHS) with kernel k and $\phi : \mathcal{X} \rightarrow \mathcal{H}$. A smaller MMD indicates a less distribution discrepancy. In this paper, we use a sum of multiple Gaussian kernels², each of which has a specific width to better capture the discrepancy.

3.2 Fine-tuning

Based on the observation that the first layer of many deep neural networks trained on images learns a function similar to Gabor filters and color blobs [15], fine-tuning assumes that the

²The list of Gaussian kernel widths used in this paper is $[1e-6, 1e-5, 1e-4, 1e-3, 0.01, 0.1, 1, 5, 10, 15, 20, 25, 30, 35, 100, 1e3, 1e4, 1e5, 1e6]$.

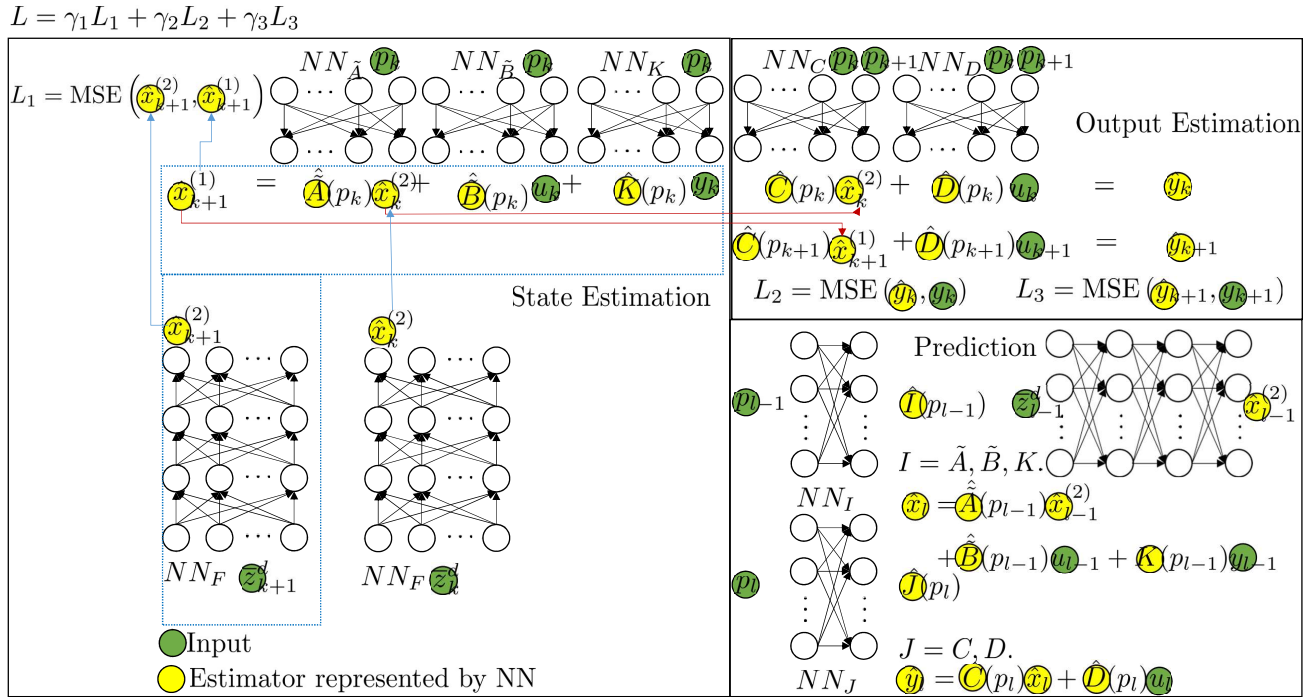


Figure 1: The complete computing graph of state integrated matrix estimation (SIME) method [13]. NN_I ($I = \tilde{A}, \tilde{B}, K, C, D, F$) are used to distinguish neural networks. The green and yellow circles represent the inputs and outputs of their adjacent neural networks, respectively (except for $x_k^{(2)}$ and $x_{k+1}^{(2)}$ which are computed by NN_F). The red lines show the connections between computing graphs of state and output estimation modules.

first few layers of deep neural networks for a task can encode knowledge that are common for other similar tasks. By reusing these layers, fine-tuning can facilitate the learning of other tasks. This approach is equivalent to assuming identical basis functions for identification of similar systems using a family of functions with affine scheduling dependency in the context of LPV model identification.

Fine-tuning design requires determining the way to share both structure and parameters of ANNs. From the perspective of function approximation, the structure of ANN determines the family of functions. According to the prior knowledge on the relative complexity of two similar systems, the original structure of the previous model can be shared for the model of a new system. If the new system is supposed to be more complex than the original system, for example, we can add more layers to the original model. From the perspective of optimization, the original parameters of the identified model can provide a good initialization for the training of the new model. Furthermore, we can copy the parameters of the first few layers of the original ANN to the new ANN, fix them and train the remaining parameters. The advantages of this approach to reducing the number of the trainable parameters are threefold: providing regularization, avoiding

over-fitting, and improving computational efficiency.

3.3 Framework of Online Transfer Learning

Using online transfer learning, we propose the workflow shown in Figure 2 to refine the existing offline model in real time. Strictly speaking, online learning updates the identified model for future data at each time step. However, considering the complex dynamics of a system and the feasibility of collecting data and refining model in a reasonable (short) amount of time, we use batch data to tune the model but restrict the computation time such that the refinement can be completed in time with the system operating normally.

In this paper, we aim to enhance the control performance by improving the accuracy of identified model, although tuning matrix weights Q and R in (5) can further boost the performance of MPC. Moreover, we assume that the range of the scheduling variables in the closed-loop batch data is identical to the scheduling variable set $\mathbb{P}$ to fine-tune the global model. The scenario where only batch data over a proper subset of $\mathbb{P}$ can be obtained will be investigated in the future work.

Moreover, we propose multiple metrics to monitor the process of online transfer learning. As the metrics are estimated

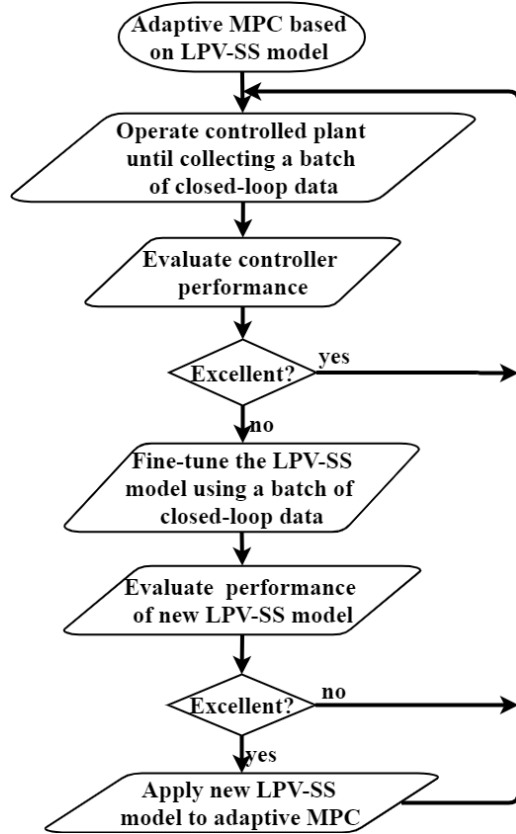


Figure 2: Workflow of refining LPV-SS model in real time.

using data, employing only one metric can result in a poor transfer strategy. Firstly, MMD in (9) will be used to measure the distribution discrepancy both between offline data and closed-loop batch data and between the present closed-loop batch data and the previous batch. This measure provides a metric for devising transfer strategy. Another metric is derived from Kalman filter, which is a commonly used approach to tackle plant-model mismatch problem [25]. The convergence of the measurement post-fit residual can reflect the accuracy of the estimated model. In this paper, we use³

$$e_{KF} = \frac{1}{N_B} \sum_{k=1}^{N_B} |y_k - \hat{C}(p_k)\hat{x}_{k|k}|, \quad (10)$$

where N_B is the number of intervals for a batch of closed-loop data, y_k is the measured output at time instant k , $\hat{C}$ is the current estimation of output matrix using fine-tuned ANN and $\hat{x}_{k|k}$ is the posterior state estimation of Kalman filter using the current estimated matrices. Additionally, the mean absolute error

³Here, we assume no feedthrough matrix D .

$e_{TP} = \frac{1}{N_B} \sum_{k=1}^{N_B} |r_k - y_k|$ is used to measure the tracking performance and the best fit rate (BFR)

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

is employed to measure the accuracy of the identified LPV-SS model.

4 Experimental Results and Validation

The developed online transfer learning method is evaluated for combustion control of an RCCI engine with the specifications listed in Table 1. The RCCI engine is based on a 4-cylinder GM Ecotec LHU engine that was modified for RCCI operation. Details of the engine setup are found in [26]. In this paper, a high-fidelity experimentally validated RCCI engine model from [20] is used to simulate RCCI indicated mean effective pressure (IMEP) and combustion phasing that is represented by CA50 as the crank angle by which 50% of fuel has been burnt in an engine cycle.

Table 1: RCCI engine specifications

Description [Unit]	Operating Value
Bore [cm] $\times$ Stroke [cm]	8.6 $\times$ 8.6
Compression ratio	9.2:1
Displacement volume [cc]	1998
Max engine power [kW@rpm]	164@5300
Max engine torque [Nm@rpm]	353@2400
IVO, IVC [CAD bTDC]	25.5, 2
EVO, EVC [CAD bTDC]	36, 22
DI fuel rail pressure [bar]	100
PFI fuel rail pressure [bar]	3

Structure of the designed RCCI engine controller is shown in Figure 3. The designed controller is based on an MPC that incorporates an LPV state-space model for RCCI engine dynamics. The MPC scheme controls IMEP and CA50 by adjusting total injected fuel (FQ) and start of injection (SOI) of the reactive fuel (i.e., n-heptane), respectively. The MPC controller uses a Kalman Filter for estimating engine states (e.g., in-cylinder gas temperature at start of combustion) that are difficult to measure.

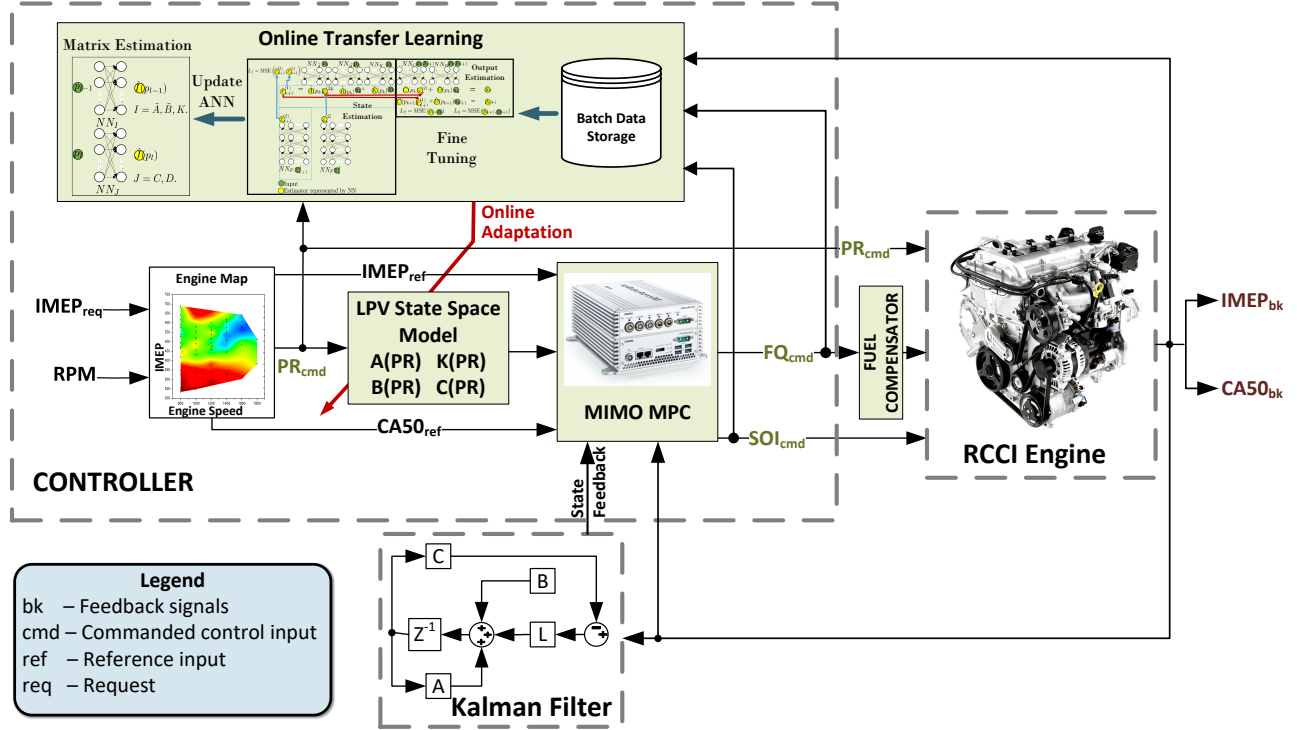


Figure 3: Schematic of the designed RCCI engine controller with online transfer learning

Here, the high-fidelity RCCI model is used to: (i) serve as our evaluation testbed, and (ii) create engine data for model identification. Using the RCCI model, two different simulation models $\mathcal{P}_1$ and $\mathcal{P}_2$ are created for the RCCI engine to validate our proposed online transfer learning method (Figure 3). These two models have different parameters but identical inputs, outputs and scheduling variable as follows

$$U = [SOI \ FQ]^T, \quad (11)$$

$$p = PR, \quad (12)$$

$$Y = [CA50 \ IMEP]^T, \quad (13)$$

where PR is the premixed ratio of dual fuels (i.e., iso-octane and n-heptane) and used as the scheduling parameter for the LPV state-space model.

The model $\mathcal{P}_1$ was identified using SIME in [13] and data $\mathcal{D}_1$ in Figure 4 (note that a periodic scheduling variable PR was used). We used the adaptive MPC described in Section 2.2 to design an initial controller for $\mathcal{P}_2$ based on the identified LPV-SS model $\mathcal{M}_1$ of $\mathcal{P}_1$. For MPC design, we assumed that Q is an identity matrix and R is an identity matrix scaled by 0.1 in (5).

The constraints on inputs and outputs are as follows

$$\begin{aligned} -35 \leq \Delta u_{SOI} \leq 35 \text{ [CAD]}, \quad -17.5 \leq \Delta u_{FQ} \leq 17.5 \text{ [mg/cycle]}, \\ 0 \leq u_{SOI} \leq 70 \text{ [CAD bTDC]}, \quad 0 \leq u_{FQ} \leq 35 \text{ [mg/cycle]}, \\ -10 \leq y_{CA50} \leq 30 \text{ [CAD aTDC]}, \quad 500 \leq y_{IMEP} \leq 1000 \text{ [kPa]}, \end{aligned}$$

which are based on the experimental data used to validate the RCCI engine dynamic model. The scale factor for $IMEP$ is 10. Moreover, the initial MPC was built based on the trained SIME model evaluated at $PR = 20$. The aim of this experiment⁴ is to refine $\mathcal{M}_1$ to obtain an improved LPV-SS model $\mathcal{M}_2$ for $\mathcal{P}_2$ and achieve a better control performance than the initial controller.

To quantify the discrepancy between $\mathcal{P}_1$ and $\mathcal{P}_2$ and satisfy the assumption that the closed-loop batch data traverse $\mathbb{P}$, we maintained the scheduling signal PR in Figure 4(a), used the output signal in Figure 4(b) as reference and the designed adaptive MPC to calculate the control inputs. In this experiment, the number of samples in a batch of closed-loop data set is 400, which corresponds to scheduling signals of two periods. Moreover, we split the batch data into training and testing sets with a ratio of 75%/25%. For fine-tuning settings, the structure of ANNs in SIME remains unchanged and the parameters in the

⁴For structure design and hyperparameter setting of SIME, refer to [13].

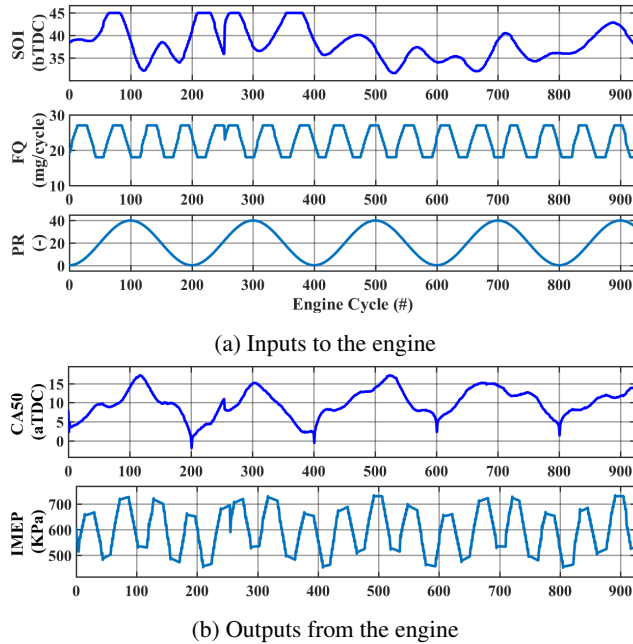


Figure 4: Inputs/outputs data set used for LPV-SS model learning of RCCI engine. The engine simulating conditions are $N = 1000$ rpm, $T_{\text{man}} = 333.15$ K and $P_{\text{man}} = 96.5$ KPa.

trained model are used for initialization. However, we did not fix any parameters due to the significant differences in the distributions of data as shown in Figure 5. To demonstrate the poor

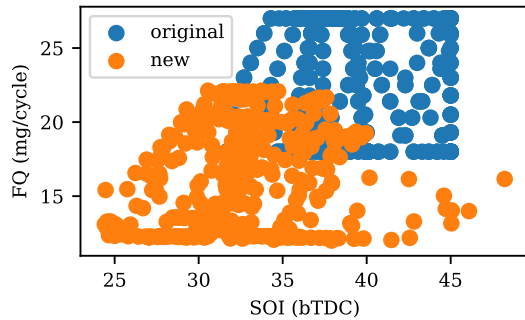


Figure 5: Distribution comparison between inputs of $\mathcal{D}_1$ and new data sets. The horizontal axis represents the first input while vertical is the second one. We note that $\widehat{\text{MMD}}(\mathcal{D}_1, \mathcal{D}_{2,1}) = 0.4215$.

prediction performance of the original model $\mathcal{M}_1$ on a batch of closed-loop data $\mathcal{D}_{2,1}$ ⁵, we directly applied that model to the new

⁵The first subscript is used to distinguish systems and the second subscript

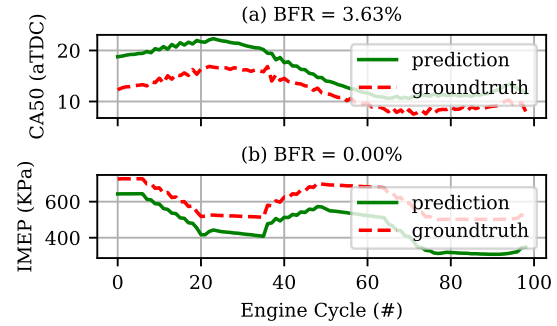


Figure 6: The outputs predicted by the original model $\mathcal{M}_1$ and the true outputs on the testing set. The engine testing conditions are $N = 1000$ rpm, $T_{\text{man}} = 363.15$ K and $P_{\text{man}} = 96.5$ KPa.

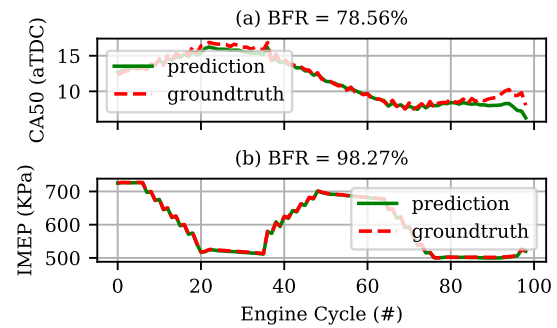


Figure 7: The outputs predicted by model $\mathcal{M}_{1,1}$ and the true outputs on the testing set. The engine testing conditions are $N = 1000$ rpm, $T_{\text{man}} = 363.15$ K and $P_{\text{man}} = 96.5$ KPa. The engine testing conditions are $N = 1000$ rpm, $T_{\text{man}} = 363.15$ K and $P_{\text{man}} = 96.5$ KPa.

data. The results show that the original model $\mathcal{M}_1$ cannot capture the dynamics of the new engine $\mathcal{P}_2$ (see Figure 6). Using $\mathcal{D}_{2,1}$ to fine-tune the model $\mathcal{M}_1$, the results in Figure 7 show that the refined model $\mathcal{M}_{1,1}$ can better capture the dynamics of new engine $\mathcal{P}_2$. Furthermore, the mean absolute tracking errors decrease from $e_{TP} = [0.8, 16.48]^T$ and $e_{KF} = [0.4287, 0.5178]^T$ to $e_{TP} = [0.6, 12.11]^T$ and $e_{KF} = [0.2843, 0.1415]^T$ in the next 1/2 period which corresponds to scheduling signals of 1/2 period. Figure 8 shows the tracking performance using model $\mathcal{M}_{1,1}$. Additionally, the fine-tuning only took 100 seconds using a computer with a 2.6 GHz CPU and 16 GB RAM, as we trained the model for only 200 epochs on 260 samples under online setting rather than 2,000 epochs on 602 samples under the offline setting.

After the engine runs for 1/2 period (which corresponds to

after the comma is the batch number.

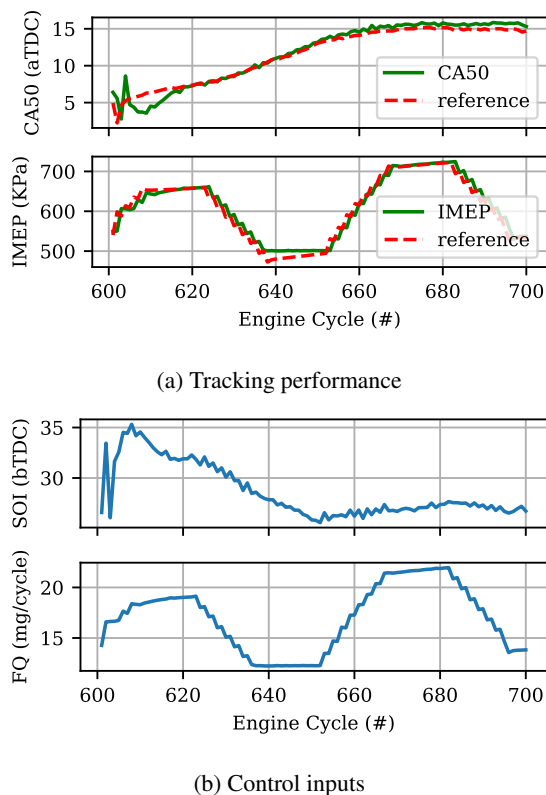


Figure 8: Tracking performance and control inputs using model $\mathcal{M}_{1,1}$. The engine simulating conditions are $N = 1000$ rpm, $T_{\text{man}} = 363.15$ K and $P_{\text{man}} = 96.5$ KPa.

the scheduling signal PR over half a period), we can combine this $1/2$ -period data in Figure 8 and the previous $3/2$ periods data to obtain a new batch of data $\mathcal{D}_{2,2}$. Now, $\overline{\text{MMD}}(\mathcal{D}_{2,1}, \mathcal{D}_{2,2}) = 0.0699$, which implies that the distribution discrepancy decreases. By sliding the time window rather than waiting for another two periods, we can update the model faster and stabilize the refinement process by using less new data. We refined $\mathcal{M}_{1,1}$ using $\mathcal{D}_{2,2}$ and obtained a new version of model $\mathcal{M}_{1,2}$ that is better than $\mathcal{M}_{1,1}$ in terms of tracking performance. The mean absolute errors decrease to $e_{TP} = [0.4, 11.42]^T$ and $e_{KF} = [0.1289, 0.0777]^T$. Figure 10 shows the tracking performance using model $\mathcal{M}_{1,2}$. Similarly, we obtain $\mathcal{D}_{2,3}$ by combining the closed-loop batch data in Figure 10 and the previous $3/2$ -period data. However, BFR_{IMEP} in Figure 9 shows that the performance of predicting IMEP slightly decreases. As e_{KF} decreases and BFR_{IMEP} is high, $\mathcal{M}_{1,2}$ is expected to be better than $\mathcal{M}_{1,1}$. We can keep refining the model until the BFR stops increasing or e_{KF} stops decreasing or e_{TP} fulfills the control requirements. Moreover, since $\overline{\text{MMD}}(\mathcal{D}_{2,2}, \mathcal{D}_{2,3}) = 0.0434$ is small, $\mathcal{D}_{2,3}$ cannot provide much information and fine-tuning stops.

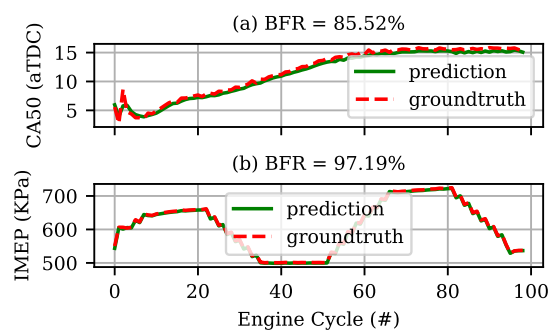


Figure 9: The outputs predicted by model $\mathcal{M}_{1,2}$ and the true outputs on the testing set. The engine testing conditions are $N = 1000$ rpm, $T_{\text{man}} = 363.15$ K and $P_{\text{man}} = 96.5$ KPa.

5 Concluding Remarks

In this paper, a framework was presented to refine identified ANN-based LPV-SS models using online transfer learning such that the closed-loop system performance using an LPV-MPC controller is enhanced. Specifically, MMD was introduced to quantify similarity, fine-tuning strategies were discussed for transfer learning with different degrees of similarity and the workflow of refining models online was provided. Experiments on two different high fidelity simulation models of an RCCI engine showed that the proposed framework can refine the offline model of one system to achieve better prediction and tracking performance on another “similar but not identical” system.

ACKNOWLEDGMENT

This work was financially supported by the United States National Science Foundation under awards #1762595 and #1912757.

REFERENCES

- [1] Bachnas, A., Tóth, R., Ludlage, J., and Mesbah, A., 2014. “A review on data-driven linear parameter-varying modeling approaches: A high-purity distillation column case study”. *Journal of Process Control*, **24**(4), pp. 272–285.
- [2] Valiant, L. G., 1984. “A theory of the learnable”. *Communications of the ACM*, **27**(11), pp. 1134–1142.
- [3] Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C., 2018. A survey on deep transfer learning.
- [4] Lazaric, A., 2012. “Transfer in reinforcement learning: a framework and a survey”. In *Reinforcement Learning*. Springer, pp. 143–173.
- [5] Wang, M., and Deng, W., 2018. “Deep visual domain adaptation: A survey”. *Neurocomputing*, **312**, pp. 135–153.
- [6] Ruder, S., Peters, M. E., Swayamdipta, S., and Wolf, T.,

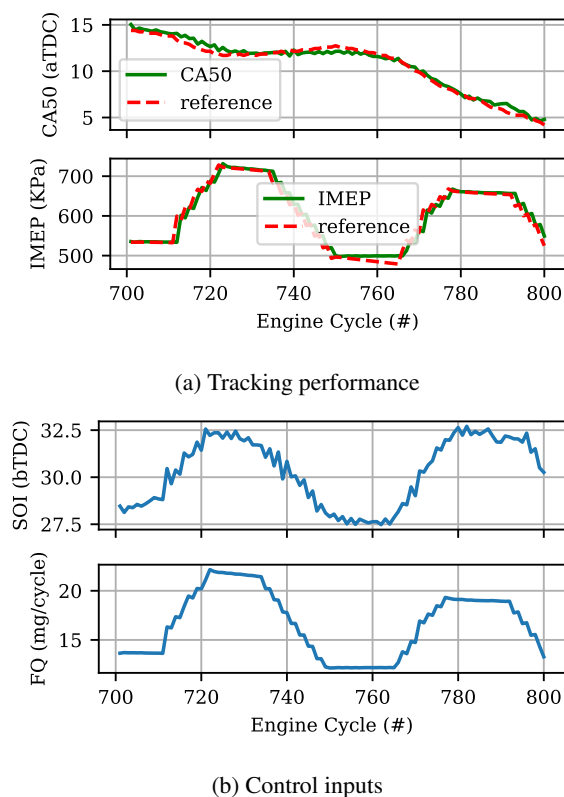


Figure 10: Tracking performance and control inputs using model $\mathcal{M}_{1,2}$. The engine simulating conditions are $N = 1000$ rpm, $T_{\text{man}} = 363.15$ K and $P_{\text{man}} = 96.5$ KPa.

2019. “Transfer learning in natural language processing”. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials, pp. 15–18.
- [7] Pan, S. J., and Yang, Q., 2009. “A survey on transfer learning”. *IEEE Transactions on knowledge and data engineering*, **22**(10), pp. 1345–1359.
- [8] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q., 2019. “A comprehensive survey on transfer learning”. *arXiv preprint arXiv:1911.02685*.
- [9] Bao, Y., Li, Y., Huang, S.-L., Zhang, L., Zheng, L., Zamir, A., and Guibas, L., 2019. “An information-theoretic approach to transferability in task transfer learning”. In 2019 IEEE International Conference on Image Processing (ICIP), IEEE, pp. 2309–2313.
- [10] Mohammadpour, J., and Scherer, C. W., 2012. *Control of linear parameter varying systems with applications*. Springer Science & Business Media.
- [11] Cox, P. B., 2018. “Towards efficient identification of linear parameter-varying state-space models”. PhD thesis, Ph. D. dissertation, Eindhoven University of Technology.
- [12] 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**, pp. 38–47.
- [13] Bao, Y., Mohammadpour Velni, J., Basina, A., and Shahbakhti, M., 2020. “Identification of state-space linear parameter-varying models using artificial neural networks”. In 21st IFAC World Congress (accepted and to appear), IFAC.
- [14] Jacot, A., Gabriel, F., and Hongler, C., 2018. “Neural tangent kernel: Convergence and generalization in neural networks”. In Advances in neural information processing systems, pp. 8571–8580.
- [15] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H., 2014. “How transferable are features in deep neural networks?”. In Advances in neural information processing systems, pp. 3320–3328.
- [16] Ansari, E., Shahbakhti, M., and Naber, J., 2018. “Optimization of performance and operational cost for a dual mode diesel-natural gas RCCI and diesel combustion engine”. *Applied Energy*, **231**, pp. 549 – 561.
- [17] Reitz, R. D., and Duraisamy, G., 2015. “Review of high efficiency and clean reactivity controlled compression ignition (RCCI) combustion in internal combustion engines”. *Progress in Energy and Combustion Science*, **46**, pp. 12–71.
- [18] 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), pp. 421–435.
- [19] Indrajuna, A., Bekdemir, C., Feru, E., and Willems, F., 2018. “Towards model-based control of RCCI-CDF mode-switching in dual fuel engines”. *SAE International*, pp. 01–0263. Paper NO. 2018-01-0263.
- [20] Raut, A., Khoshbakht Irdmoussa, B., and Shahbakhti, M., 2018. “Dynamic modeling and model predictive control of an RCCI engine”. *Control Engineering Practice*, **81**, pp. 129–144.
- [21] Sui, W., Pulpeiro González, J., and Hall, C. M., 2018. “Modeling and Control of Combustion Phasing in Dual-Fuel Compression Ignition Engines”. *Journal of Engineering for Gas Turbines and Power*, **141**(5), 11. 051005.
- [22] Irdmoussa, B. K., Rizvi, S. Z., Mohammadpour Velni, J., Nabert, J. D., and Shahbakhti, M., 2019. “Data-driven Modeling and Predictive Control of Combustion Phasing for RCCI Engines”. In 2019 American Control Conference (ACC), pp. 1617–1622.
- [23] Basina, L. N. A., Irdmoussa, B. K., Mohammadpour Velni, J., Borhan, H., Naber, J. D., and Shahbakhti, M., 2020. “Data-driven Modeling and Adaptive Control of Maximum Pressure Rise Rate in RCCI Engines”. *IEEE Conference on Control Technology and Applications*. To appear.

- [24] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A., 2012. “A kernel two-sample test”. *Journal of Machine Learning Research*, **13**(Mar), pp. 723–773.
- [25] Simkoff, J. M., Wang, S., Baldea, M., Chiang, L. H., Castillo, I., Bindlish, R., and Stanley, D. B., 2018. “Plant–model mismatch estimation from closed-loop data for state-space model predictive control”. *Industrial & Engineering Chemistry Research*, **57**(10), pp. 3732–3741.
- [26] Arora, J. K., and Shahbakhti, M., 2017. “Real-time closed-loop control of a light-duty RCCI engine during transient operations”. *SAE International*. Paper No. 2017-01-0767.