



# Deep learning based model predictive control for compression ignition engines

Armin Norouzi <sup>a,\*</sup>, Saeid Shahpour <sup>a</sup>, David Gordon <sup>a</sup>, Alexander Winkler <sup>b</sup>, Eugen Nuss <sup>c</sup>, Dirk Abel <sup>c</sup>, Jakob Andert <sup>b</sup>, Mahdi Shahbakhti <sup>a</sup>, Charles Robert Koch <sup>a</sup>

<sup>a</sup> Mechanical Engineering Department, University of Alberta, Edmonton, Canada

<sup>b</sup> Teaching and Research Area Mechatronics in Mobile Propulsion, RWTH Aachen University, Germany

<sup>c</sup> Institute of Automatic Control, RWTH Aachen University, Germany

## ARTICLE INFO

### Keywords:

Deep learning  
Machine learning  
Nonlinear model predictive Control  
Imitation controller  
Internal combustion engine  
Emission reduction

## ABSTRACT

Machine learning (ML) and a nonlinear model predictive controller (NMPC) are used in this paper to minimize the emissions and fuel consumption of a compression ignition engine. In this work machine learning is applied in two methods. In the first application, ML is used to identify a model for implementation in model predictive control optimization problems. In the second application, ML is used as a replacement of the NMPC where the ML controller learns the optimal control action by imitating or mimicking the behavior of the model predictive controller. In this study, a deep recurrent neural network including long–short term memory (LSTM) layers are used to model the emissions and performance of an industrial 4.5 liter 4-cylinder Cummins diesel engine. This model is then used for model predictive controller implementation. Then, a deep learning scheme is deployed to clone the behavior of the developed controller. In the LSTM integration, a novel scheme is used by augmenting hidden and cell states of the network in an NMPC optimization problem. The developed LSTM-NMPC and the imitative NMPC are compared with the Cummins calibrated Engine Control Unit (ECU) model in an experimentally validated engine simulation platform. Results show a significant reduction in Nitrogen Oxides (NO<sub>x</sub>) emissions and a slight decrease in the injected fuel quantity while maintaining the same load. In addition, the imitative NMPC has a similar performance as the NMPC but with a two orders of magnitude reduction of the computation time.

## 1. Introduction

Internal Combustion Engines (ICEs) are ubiquitous and they power small devices such as lawn mowers all the way to large ship engines (Norouzi, Heidarifar, Shahbakhti, Koch, and Borhan, 2021). The durability and reliability of ICEs has made them very attractive. However, the broad use of the ICE can be attributed to more than 20% of the total Greenhouse Gas (GHG) emissions worldwide (Norouzi, Aliramezani and Koch, 2021; Norouzi, Gordon, Aliramezani, & Koch, 2020). Another area of concern of ICE powered vehicles is the difference between emissions measurements taken at the test bench and the emissions that are actually emitted from tailpipe of vehicles on the road. This has resulted in new emissions legislation which now focuses on the Real Driving Emissions (RDE) in addition to test bench testing. This has led to increasing challenges for ICEs to meet current and upcoming legislation (Norouzi, Heidarifar et al., 2021). These tighter emissions and fuel economy legislation, in combination with the complexity of the combustion process, have led to the requirement for significantly more advanced engine controllers than are currently used.

New progress for ECU hardware has enabled significant improvements in the computational power of micro controllers. This increase in computational power has enabled online optimization methods to further reduce ICE emissions (Norouzi, Heidarifar et al., 2021).

Historically, feed-forward control and feedback based controllers have been used for the control of ICE's in automotive applications. The most common feed-forward controller is the use of two-dimensional look-up tables or so-called calibration maps. Where the controller is calibrated for a variety of operating points on the test bench (Guzzella & Onder, 2009; Isermann, 2014). With the addition of a feedback controller such as a Proportional Integral Derivative (PID) controller it is possible to correct for disturbances and parameter variations (Guzzella & Onder, 2009). The gains of the PID controller are first tuned using the procedure of parameter optimization while fine tuning is completed using the trial-and-error method. To comply with fuel consumption and emissions regulations, a significant number of control inputs for a broad range of engine speed and load conditions must be evaluated. This leads to a very time-consuming and costly calibration process for the engine manufacturer.

\* Corresponding author.

E-mail address: [norouzi@ualberta.ca](mailto:norouzi@ualberta.ca) (A. Norouzi).

## Nomenclature

### Abbreviations

NO <sub>x</sub>	Nitrogen Oxides
aTDC	after Top Dead Center
BM	Benchmark
CAD	Crank Angle Degree
DPM	Detailed Physical Model
ECU	Engine Control Unit
ESM	Engine Simulation Model
FC	Fully Connected
FQ	Fuel Quantity
GA	Genetic Algorithm
HPIPM	High-Performance Interior-Point Method
ICE	Internal Combustion Engine
LSTM	Long-Short Term Memory
ML	Machine Learning
MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
OC	Optimal Control Problem
PID	Proportional Integral Derivative
QP	Quadratic Programming
RNN	Recurrent Neural Network
SOI	Start Of Injection
SQP	Sequential Quadratic Programming
VGT	Variable-Geometry Turbocharger

### Mathematical symbols

$\hat{u}(k)$	Estimated optimal control input
$\hat{y}(k)$	Estimated plant output
$c(k)$	Cell state
$f(k)$	Forgot gate
$g(k)$	Cell candidate
$h(k)$	Hidden state
$i(k)$	Input gate
$o(k)$	Output gate
$P$	Horizon length
$P_{\text{man}}(k)$	Intake manifold pressure
$T_{\text{out}}(k)$	Engine-output torque
$u(k)$	Input vector
$w_{\Delta u}$	MPC weight for engine-output torque
$w_{\text{FQ}}$	MPC weight for fuel consumption minimization
$w_{\text{NO}_x}$	MPC weight for NO <sub>x</sub> minimization
$W_{\text{FC}}$	Weight matrices of fully connected layer
$W_{h(f,g,i,o)}$	Weight matrices to hidden state vector $h(k)$
$w_s$	MPC weight for Constraint violation penalty
$w_{T_{\text{out}}}$	MPC weight for engine-output torque
$W_{u(f,g,i,o)}$	Weight matrices to input vector $u(k)$
$x(k)$	State vector
$y(k)$	Output vector
$z(k)$	Internal output of each layer

One solution to this problem is use of model-based feedback controllers. Several model-based controllers have been used in engine feedback control over the last five decades (Powell, 1987) including Linear Quadratic Regulator (LQR) (López, Espinosa, & Agudelo, 2011),

Linear Quadratic Gaussian (LQG) (Pfeiffer et al., 2004), Sliding Model Control (SMC) (Norouzi, Ebrahimi, & Koch, 2019), Adaptive (Souder & Hedrick, 2004), and Model Predictive Control (MPC) (Basina et al., 2020; Irdmoussa, Rizvi, Velni, Naber, & Shahbakhti, 2019). Each of these model-based controllers has its own advantages and disadvantages; however, MPC is one of the most promising control methods since it has the ability to control highly constrained nonlinear systems. MPC is ideal for ICEs since: (i) ICE combustion is significantly nonlinear, (ii) all system inputs are generally constrained based on physical limitations, and (iii) ICE deals with multi-objective optimization including fuel economy, emissions, noise, and operator satisfaction (i.e., load following during engine transients). Additionally, MPC can provide an optimal real-time solution for meeting the multi-objective goals of time-critical systems (Basina et al., 2020; Irdmoussa et al., 2019; Winkler et al., 2021).

Beginning in 1998, MPC was applied to ICE control for the first time in a simulation environment for the air-fuel ratio (AFR) control of an spark ignition (SI) gasoline engine using a neural network based linear AFR model (Lennox, Montague, Frith, & Beaumont, 1998). Then in 1999, using a GT-Power engine model simulations, the idle speed control of an SI engine was explored using MPC (Bromnick, 1999). These early works were done in simulation environments, while recent works (Broomhead, Manzie, Hield, Shekhar, & Brear, 2016; Liao-McPherson et al., 2020; Saerens, Diehl, Swevers, & Van den Bulck, 2008; Stewart & Borrelli, 2008; Yin, Turesson, Tunestål, & Johansson, 2020) include experimental implementation of nonlinear multi-objective MPC on real engines. MPC has been used in ICE control for a wide variety of applications. Among these, MPC has been used widely for control of fuel consumption (Asprion, Chinellato, & Guzzella, 2014; Bemporad, Bernardini, Long, & Verdejo, 2018; Karlsson, Ekholm, Strandh, Johansson, & Tunestål, 2010; Zhao et al., 2014; Zhu, Prucka, Prucka, & Dourra, 2017), combustion phasing (Basina et al., 2020; Irdmoussa et al., 2019; Karlsson et al., 2010; Zhu et al., 2017), cyclic variability (Zhu et al., 2017), torque and load (Basina et al., 2020; Bemporad et al., 2018; Karlsson et al., 2010; Nuss et al., 2019; Zhu et al., 2017), idle speed (Bromnick, 1999), intake airpath control (Dahl et al., 2018; Herczeg, Raff, Findeisen, & Allgöwe, 2006; Huang, Liao-McPherson, Kim, Butts, & Kolmanovsky, 2018; Yashiro et al., 2017; Zhao et al., 2014; Zhu et al., 2017), knock and Maximum Pressure Rise Rate (MPRR) (Basina et al., 2020; Zhu et al., 2017), engine-out emissions (Karlsson et al., 2010; Liao-McPherson et al., 2020; Liu, Dizqah, Herreros, Schaub, & Haas, 2021; Zhao et al., 2014), and exhaust aftertreatment (Dahl et al., 2018).

MPC is a promising control technique which has been gaining significant attention due to the following five main advantages: (1) implicitly considers constraints on state, input and output variables, (2) provides closed loop control performance and stability for the optimal problem with constraints, (3) exploits the use of a future horizon while optimizing the current control law, (4) offers the possibility of both offline and real-time implementations, and (5) provides the capability to handle uncertainty in system's parameters, delays, and non-linearity in the model (Norouzi, Heidarifar et al., 2021). However, like other model based control strategies, the challenge with MPC is the requirement for an accurate plant model and the increased computational effort that is required to evaluate the model over the prediction horizon. These two aspects can work against each other as increasingly complicated models are introduced to minimize model/plant mismatch resulting in a significant increase in computation (Bemporad, 2020).

To solve these challenges of MPC, the integration of ML and MPC has been offered as an emerging area that offers the potential to enhance the control performance of MPC. Combining ML and MPC first began in the early 2000s but has since accelerated significantly in the last five years. ML integration with MPC, sometimes referred to as learning MPC, is divided into three main categories of: (1) ML in modeling, (2) ML in computational reduction, and (3) MPC in safe learning (Norouzi, Heidarifar et al., 2021). ML in modeling is used to

create a high fidelity model of the plant to be used inside MPC. ML in computational reduction is aimed at reducing the high computational costs associated with the optimization in MPC. One method used to reduce the computational time of MPC is cloning the behavior of an MPC using ML. In this case, the online MPC is replaced by an ML model of the controller (Hewing, Wabersich, Menner, & Zeilinger, 2020; Norouzi, Heidarifar et al., 2021). In safe learning, an ML learning controller is used but MPC is used to ensure that constraints are satisfied and system stability can be guaranteed.

Physics-based modeling provides accurate models based on physical insight into the system, but the engine combustion and emissions models utilize detailed 3D models which are computationally expensive (Gordon et al., 2021; Shahpour et al., 2021a). This makes physical models impractical for real-time model-based control. Current production ECUs are not capable of running these detailed physical models and thus these models cannot be implemented for real-time control. An alternative approach for engine and emission modeling is ML based modeling which uses measurement data to train a data-driven model. These models can be as accurate as physical models while requiring significantly less computational time. This is desired for implementation of the model-based controllers in ECUs (Shahpour et al., 2021a). Using ML in modeling has been successfully implemented for diesel engine using support vector machine (SVM) (Basina et al., 2020; Irdmoussa et al., 2019; Khoshbakht Irdmoussa, Naber, Mohammadpour Velni, Borhan, & Shahbakhti, 2021), neural network (Hu, Chen, Wang, Chen, & Ren, 2018; Lennox et al., 1998), and Extreme Learning Machine (ELM) (Janakiraman, Nguyen, & Assanis, 2016; Vaughan & Bohac, 2015). One of the methods explored in this study is using a Deep Recurrent Neural Network such as a Long-Short Term Memory (LSTM) network.

A Recurrent Neural Network (RNN) is structurally similar to a feedforward neural network with the exception of backward connections used to handle sequential relations. The advantage of the RNN compared to conventional feedforward neural network for dynamic modeling is its computational efficiency which is the result of parameter sharing. However, conventional RNN cannot accurately capture any long-term dependencies of the model as the prediction can be described as utilizing a “vanishing gradient”. This results in the contribution of earlier steps becomes increasingly small. To solve this long term memory problem of RNN, various types of cells with long-term memory have been introduced. The most popular and well-known of these long-term memory cells is LSTM (Géron, 2019). Therefore, a deep network with LSTM layer is capable of predicting the dynamics of a system with high accuracy. To the author’s knowledge, a deep network with LSTM layer has not been used for emission modeling for MPC in ICE control problem to date.

Even with ML-based modeling, MPC computational times (especially for deep networks) is high. Additionally, depending on the complexity of the control problem, convexity, and dynamic time scale it might not be feasible for real-time implementation. ICEs are complex nonlinear systems to control and therefore high computational effort is required for real-time implementation. One method to reduce the MPC computational load is to replace the MPC with an ML controller. The ML is trained to mimic the MPC controller’s behavior to achieve a significant computational time reduction for real-time implementation. For the original implementation the optimization is done using a powerful prototype ECU. Using this online implementation the ML function can be trained with data collected from the online MPC output. This ML function mimics the output of the online MPC, however, it can be implemented with a significant reduction in the computational time compared to the online MPC. Here, to reduce the computational time of the MPC, an ML based imitation of the MPC controller is proposed where a deep network is trained using the MPC inputs and outputs. Imitation MPC has been previously explored in the control of heating, ventilation, and air conditioning (HVAC) systems (Dr̄goňa, Picard, Kvasnica, & Helsen, 2018; Karg & Lucia, 2018),

vehicle dynamics control (Sun, Peng, Zhan, & Tomizuka, 2018; Zhang, Bujarbaruah, & Borrelli, 2019), robotics (Cera & Agogino, 2018), and power conversion (Novak & Dragicevic, 2020) industries and has shown great success. The application of imitation MPC to ICE control has also explored in the literature in terms of approximate MPC (Moriyasu, Nojiri, Matsunaga, Nakamura, & Jimbo, 2019; Moriyasu et al., 2018) and inverse model controller (Peng, Huo, & Hall, 2022). However, the application of deep RNN especially using LSTM in imitation MPC to ICE control is lacking in the literature.

The main contributions of this paper are

#### 1. Machine Learning based modeling for MPC design:

- Developed a transient engine performance and emission model based on Long-Short Term Memory (LSTM) that is capable of providing a high accuracy model for nonlinear model predictive control,
- Developed a novel approach to augment LSTM in NMPC problems (LSTM-NMPC) by augmenting LSTM hidden and cell state into a nonlinear optimization problem,
- Designed an NMPC based on an ML model to minimize engine-out emission and fuel consumption while maintaining the same output torque performance when compared with a benchmark model, i.e., Cummins calibrated ECU-based Engine Simulation Model (ESM).
- Compared acados, Matlab fmincon, and EMBOTECH FORCES PRO solvers and verified real-time implementation of acados in Processor-in-the-loop (PIL) setup

#### 2. Machine Learning based control to reduce the computational time of MPC:

- Developed an imitation based controller using deep neural network to clone the behavior of LSTM-NMPC to reduce computational time of optimization with same NMPC performance

The structure of the paper is shown in Fig. 1 and it includes logging data, data-driven model development, and controller implementation using ESM co-simulation. The first step including experimental data collection and detailed physical modeling using GT-power was done in our previous papers (Shahpour et al., 2021a, 2021b). Then, randomly generated inputs are fed into the GT-power model and the output data is recorded for modeling. Using these input–output pairs of data, the LSTM model can then be developed. This model is used for the design of a NMPC controller. Finally, this NMPC controller is used to train the ML based imitation controllers. All of the controllers developed in this study are simulated using an ESM co-simulation.

## 2. Engine Simulation Model (ESM)

The engine platform used in this paper is a 4.5-liter medium-duty Cummins diesel engine. Physical modeling of this engine is carried out using the GT Power software, which contains chemical and gas exchange sub-models that simulate the complex combustion processes. The proposed control strategies are tested in simulation using an Engine Simulation Model (ESM) parameterized to the real engine (Shahpour et al., 2021a, 2021b). The combustion model is calibrated using experimental data where approximately 15% of the raw data is used to calibrate the combustion model using the Genetic Algorithm (GA) algorithm in the GT-suite© software. The calibration process uses NSGA-III (Deb & Jain, 2013) for multi-objective Pareto optimization as the search algorithm. The DIPulse model is employed as the combustion model because it can deal with multi-injection combustion engines. An extended Zeldovich  $\text{NO}_x$  model is added to the combustion model in order to add a  $\text{NO}_x$  prediction to the ESM. Additional details regarding the model structure and accuracy of the ESM can be found in Shahpour et al. (2021a, 2021b). This model has an accuracy of  $\pm 5.8\%$ ,  $\pm 4.6\%$ ,

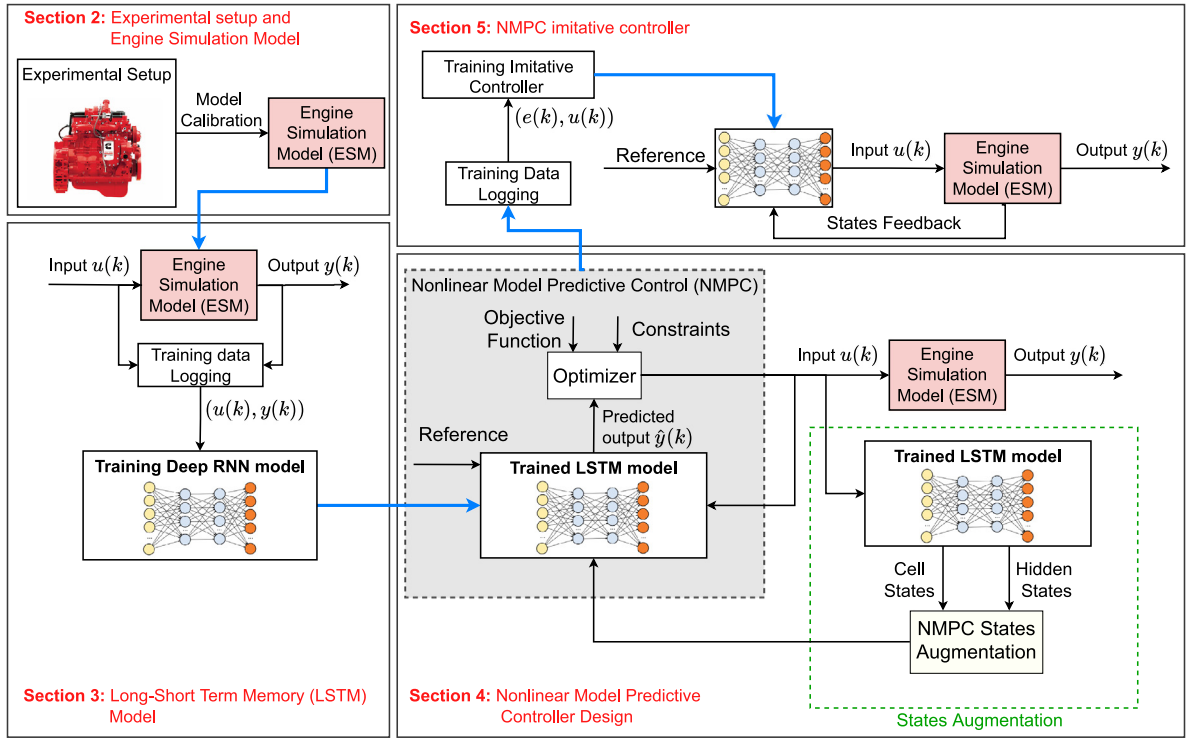


Fig. 1. Modeling and controller design procedure based on Engine Simulation Model (ESM).

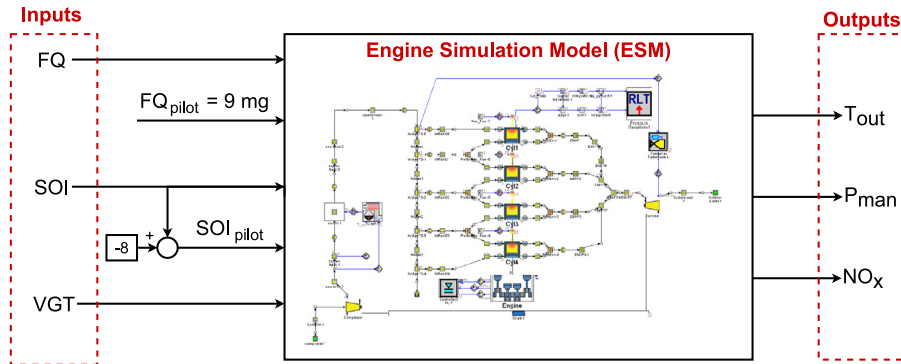


Fig. 2. Schematics of Engine Simulation Model (ESM).

and  $\pm 18.1\%$  in prediction of maximum in-cylinder pressure, intake manifold pressure, and  $\text{NO}_x$ . This ESM is used to develop data-driven models, design controllers, and evaluate the developed controllers.

The main inputs and outputs of this ESM are shown schematically in Fig. 2. The inputs of this model are Fuel Quantity (FQ), Start Of Injection (SOI) for main injection, and Variable Geometry Turbine (VGT) rate. For simplicity, the pilot injection is kept constant at 9 mg of injected fuel per cycle. The pilot injection timing varies with the main SOI, however, is kept at a constant 8 Crank Angle Degree (CAD) before the main injection. By adding these constraints between inputs it was possible to reduce the control action from 5 manipulated variables to 3. The outputs of the ESM are output torque ( $T_{\text{out}}$ ), intake manifold pressure  $P_{\text{man}}$ , and  $\text{NO}_x$  emissions.

### 3. Long-Short Term Memory network (LSTM) model

LSTM is one of the most popular and well-known variants of Recurrent Neural Networks (RNN). In comparison to RNN, LSTM utilizes a hidden state that is split into two main parts as shown schematically in Fig. 3:  $h(k)$  the short-term state, and  $c(k)$  the long-term state. The

long-term state  $c(k-1)$  travels through the network and first enters the forget gate  $f(k)$  where past values are dropped. Then additional values (memories) are added to the input gate  $i(k)$  at each time step. Therefore at each time step some data is added, and some is dropped. Further, after adding new memory, the long-term state is replicated, passed into the hyperbolic tangent activation function ( $\tanh$ ), and the output gate filters the result to generate the short-term state  $h(k)$  (equal to the cell's output  $y(k)$  for this time step) (Géron, 2019). LSTM computations can be summarized as:

$$\begin{aligned}
 i(k) &= \sigma(W_{ui}^T u(k) + W_{hi}^T h(k-1) + b_i) \\
 f(k) &= \sigma(W_{uf}^T u(k) + W_{hf}^T h(k-1) + b_f) \\
 g(k) &= \tanh(W_{ug}^T u(k) + W_{hg}^T h(k-1) + b_g) \\
 o(k) &= \sigma(W_{uo}^T u(k) + W_{ho}^T h(k-1) + b_o) \\
 c(k) &= f(k) \odot c(k-1) + i(k) \odot g(k) \\
 h(k) &= y(k) = o(k) \odot \tanh(c(k))
 \end{aligned} \tag{1}$$

where  $W_{u(f,g,i,o)}$  and  $W_{h(f,g,i,o)}$  are the weight matrices to input vector  $u(k)$  and previous short-term state  $h(k)$ . In this equation,  $\odot$ , is an



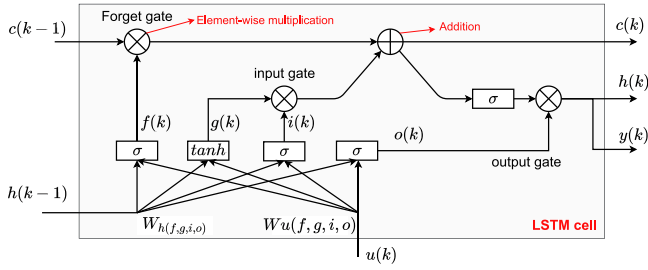


Fig. 3. Long-Short Term Memory (LSTM) cell structure.

element-wise multiplication and  $b_{(f,g,i,o)}$  are the biases. In Eq. (1),  $i(k)$ ,  $f(k)$ ,  $g(k)$ ,  $o(k)$ ,  $c(k)$ , and  $h(k)$  are input gate, forget gate, cell candidate, output gate, cell state, and hidden state, respectively. The activation functions used in LSTM computation are  $\tanh(z)$  and  $\sigma(z)$  which are defined as

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2)$$

$$\tanh(z) = \frac{e^{2z} - 1}{e^{2z} + 1} \quad (3)$$

A 2-level deep network, including two series networks, one for predicting intake manifold pressure and output torque and another for  $\text{NO}_x$  emissions is proposed. Each network contains one LSTM layer and three fully connected (FC) layers. The structure of this network is shown schematically in Fig. 4. The main reason for having two separate networks is a physical understanding of the system. As  $\text{NO}_x$  is created during combustion and usually measured through a sensor after each combustion cycle, it depends not only on  $u(k)$  but also on intermediate states such as intake manifold pressure. For modeling  $\text{NO}_x$ , having output torque is also helpful and adds more features for improving prediction accuracy. Based on a physical understanding of the system,  $\text{NO}_x$  depends on all five features, but the other two outputs, including output torque and intake manifold pressure, are not dependent on  $\text{NO}_x$ . As LSTM is used in this architecture and has a recurrent behavior, adding all outputs in a single network makes intake manifold pressure and output torque a function of  $\text{NO}_x$ , which is incorrect from a physical point of view.

The FC layer is added around the LSTM layer to increase the network capacity to better estimate the nonlinearity of the engine emissions and performance without increasing the number of hidden and cell states. Increasing the number of LSTM layers increases the number of hidden states which makes the subsequent NMPC problem larger. A fully connected equation with ReLU activation function is defined as

$$z_{FC}(k) = \text{ReLU}(W_{FC}^T u(k) + b_{FC}) \quad (4)$$

where ReLU is a Rectified Linear Unit (ReLU) activation function which is defined as

$$\text{ReLU}(z) = \begin{cases} 0 & \text{if } z \leq 0 \\ z & \text{if } z > 0 \end{cases} \quad (5)$$

As state and output functions are in the form of  $x(k+1) = F(x(k), u(k))$  and  $y(k) = F(x(k), u(k))$  need to be imported in the NMPC framework, the equation used as a predictive model of MPC is presented here. To generate state and output functions, the forward propagation needs to be evaluated for the proposed network in Fig. 4. The forward propagation of the first part of the proposed network is used to estimate engine-out torque and intake manifold pressure are given as

$$z_{FC1}(k) = \text{ReLU}(W_{FC1}^T u(k) + b_{FC1}) \quad (6a)$$

$$h_{LSTM1}(k) = \underbrace{o_{LSTM1}(k) \odot \tanh(c_{LSTM1}(k))}_{\text{based on Eq. (1)}} \quad (6b)$$

$$z_{FC2}(k) = \text{ReLU}(W_{FC2}^T h_{LSTM1}(k) + b_{FC2}) \quad (6c)$$

$$\underbrace{z_{FC3}(k)}_{[T_{out}(k) \ P_{man}(k)]^T} = \text{ReLU}(W_{FC3}^T z_{FC2}(k) + b_{FC3}) \quad (6d)$$

Each of the above equations refers to the output of each layer in the left part of Fig. 4 where  $u(k)$  are the system inputs which are defined as

$$u(k) = [\text{FQ}(k) \ \text{SOI}(k) \ \text{VGT}(k)]^T \quad (7)$$

The estimated intake manifold pressure  $P_{man}$  and output torque  $T_{out}$  are then augmented with system inputs  $u(k)$  to estimate the engine-out  $\text{NO}_x$  emissions as

$$z_{FC4}(k) = \text{ReLU}(W_{FC4}^T \begin{bmatrix} u(k) \\ \underbrace{z_{FC3}(k)}_{[T_{out}(k) \ P_{man}(k)]^T} \end{bmatrix} + b_{FC4}) \quad (8a)$$

$$h_{LSTM2}(k) = \underbrace{o_{LSTM2}(k) \odot \tanh(c_{LSTM2}(k))}_{\text{based on Eq. (1)}} \quad (8b)$$

$$z_{FC5}(k) = \text{ReLU}(W_{FC5}^T h_{LSTM2}(k) + b_{FC5}) \quad (8c)$$

$$\underbrace{z_{FC6}(k)}_{\text{NO}_x(k)} = \text{ReLU}(W_{FC6}^T z_{FC5}(k) + b_{FC6}) \quad (8d)$$

The output of this network can be calculated as

$$\hat{Y}(k) = \underbrace{[T_{out}(k) \ P_{man}(k)]}_{z_{FC3}(k)} \underbrace{[\text{NO}_x(k)]}_{z_{FC6}(k)} = \begin{bmatrix} z_{FC3}(k) \\ z_{FC6}(k) \end{bmatrix} \quad (9)$$

The augmented states of this model with hidden states and cell state to use inside NMPC are

$$x(k) = \begin{bmatrix} h_{LSTM1}(k) \in \mathbb{R}^{26} \\ h_{LSTM2}(k) \in \mathbb{R}^{26} \\ c_{LSTM1}(k) \in \mathbb{R}^{26} \\ c_{LSTM2}(k) \in \mathbb{R}^{26} \end{bmatrix} \in \mathbb{R}^{104} \quad (10)$$

The cost function for this network is as

$$J(W, b) = \frac{1}{m} \sum_{k=1}^m \mathcal{L}(\hat{Y}(k), Y(k)) + \frac{\lambda}{2m} \sum_{l=1}^L \|W^{[l]}\|_2^2 \quad (11)$$

where  $\mathcal{L}(\hat{Y}(k), Y(k))$  is the loss function,  $\lambda$  is the regularization coefficient, and  $\|W^{[l]}\|_2^2$  is the Euclidean norm which is defined as

$$\|W^{[l]}\|_2^2 = \sum_{i=1}^{n^{[l]}} \sum_{j=1}^{n^{[l-1]}} (w_{ij}^{[l]})^2 \quad (12)$$

The Mean Squared Error (MSE) cost function is used and it is defined as

$$\mathcal{L}(\hat{Y}(k), Y(k)) = \frac{1}{m} \sum_{k=1}^m (\hat{Y}(k) - Y(k))^2 \quad (13)$$

The training information along with the design values for the proposed network are summarized in Table 1. To train this model, MATLAB Deep Learning Toolbox© utilizing the Adam algorithm has been used in this work. In Fig. 5, the loss function versus iteration for both the performance and emission networks are shown, where within a defined number of Epochs, the loss functions converges to a minimum. The validation loss function also converges to match the training function, indicating that no overfitting or underfitting of the model has occurred.

Fig. 6 shows the training and validation results of the proposed model. To develop this neural network based model, which has more than 11,000 learnable parameters, a larger data set was required. Therefore, the ESM was run for 100,000 engine cycles. Referring to Fig. 6, cycles 1 to 80,000 are devoted for training and cycles 80,001 to

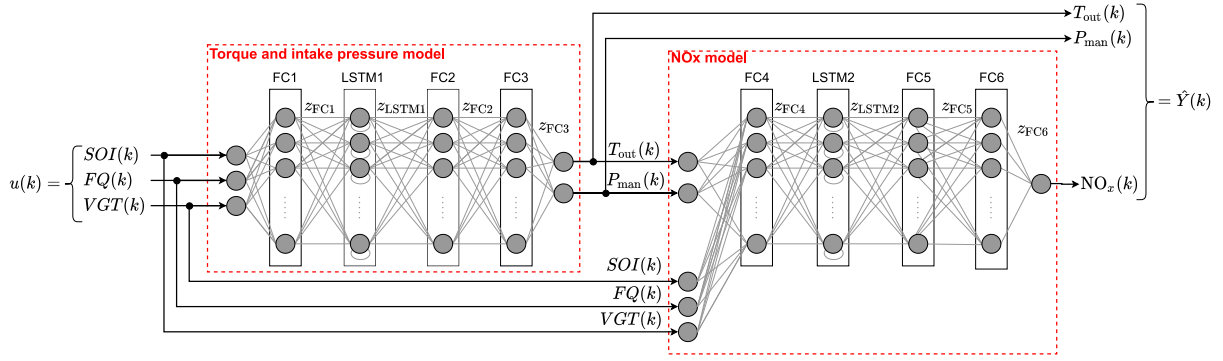


Fig. 4. Structure of proposed deep neural network model for engine performance and emission modeling. FC: Fully connected layer, LSTM: Long-short term memory.

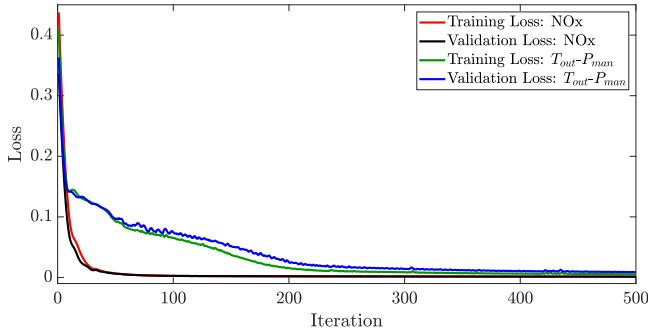


Fig. 5. Loss vs. Iteration for NO<sub>x</sub>, torque and pressure model.

Table 1

Properties of 2-level engine performance and emission LSTM-based model.

Name	Value
FC(1,2,3,4,5,6) size	26
LSTM(1,2) size	26
Optimizer	Adam
Maximum Epochs	500
Mini batch size	512
Learn rate drop period	150 Epochs
Learn rate drop factor	0.5
L2 Regularization	0.1
Initial learning rate	0.001
Validation frequency	1
Momentum	0.9
Squared gradient decay	0.99

100,000 are used for validation. The LSTM model can estimate intake manifold pressure, output torque, and NO<sub>x</sub> emission with high accuracy for both training and validation data sets. The accuracy of the training data is 2.35%, 1.98%, and 1.07% for NO<sub>x</sub>,  $T_{out}$ , and  $P_{man}$ , respectively. For the validation data set, an accuracy of 2.86%, 2.27%, and 1.53% for NO<sub>x</sub>,  $T_{out}$ , and  $P_{man}$  are found.

As this developed model will be used inside a Nonlinear Model Predictive Controller (NMPC) the accuracy of these models is critical. To compare the models, the LSTM model is run simultaneously in one simulation where the prediction of the LSTM models is compared against the ESM co-simulation. In this manner, the LSTM model is evaluated using newly generated test data that is new for the models. Fig. 7 shows the model comparison where the LSTM model is capable of estimating all outputs to a high accuracy.

To summarize the accuracy of the model, the Normalized Root Means Square Error (NRMSE) between the ESM and LSTM model are calculated. The RMSE is normalized using the range (defined as the maximum value minus the minimum value) of the ESM logged data. The NRMSE for NO<sub>x</sub>,  $P_{man}$ , and  $T_{out}$  are 4.04%, 2.21%, and 2.35%,

respectively. As shown, this model is capable of predicting emission and output torque with an error of less than 5%.

#### 4. Nonlinear model predictive controller design

Model predictive control (MPC) uses the principle of receding horizons in order to solve an optimization problem at each sampling instance to determine the control input. Here the objective of the controller is to minimize engine-out NO<sub>x</sub> emissions and fuel consumption while meeting the requested engine-output torque. The cost function  $J(u(\cdot|k), s(k))$  of the finite horizon optimal control problem (OCP) with horizon length  $p$  is defined as

$$J(u(\cdot|k), s(k)) = \sum_{i=0}^{p-1} \left[ \underbrace{\|T_{out}(k+i) - T_{out, ref}(k+i)\|_{w_{T_{out}}}^2}_{\text{Torque output tracking}} + \underbrace{\|NO_x(k+i)\|_{w_{NO_x}}^2}_{\text{NO}_x \text{ minimization}} + \underbrace{\|FQ(k+i)\|_{w_{FQ}}^2}_{\text{fuel consumption minimization}} + \underbrace{\|u(k+i) - u(k+i-1)\|_{w_{\Delta u}}^2}_{\text{control effort penalty}} + \underbrace{w_s s(k)^2}_{\text{Constraint violation penalty}} \right] \quad (14)$$

where

$$\| \cdot \|_w^2 = [\cdot]^T w [\cdot] \quad (15)$$

and  $s(k)$  is a nonnegative slack variable used to penalize the MPC cost function in the worst-case constraint violation,  $w_v, v \in [T_{out}, NO_x, FQ, \Delta u, s]$ , are the MPC weights, and  $z_k^T$  is the optimization decision defined as

$$u(\cdot|k) = [u(k|k)^T \ u(k+1|k)^T \ \dots \ u(k+p-1|k)^T] \quad (16)$$

The model's outputs and manipulated variables are defined as

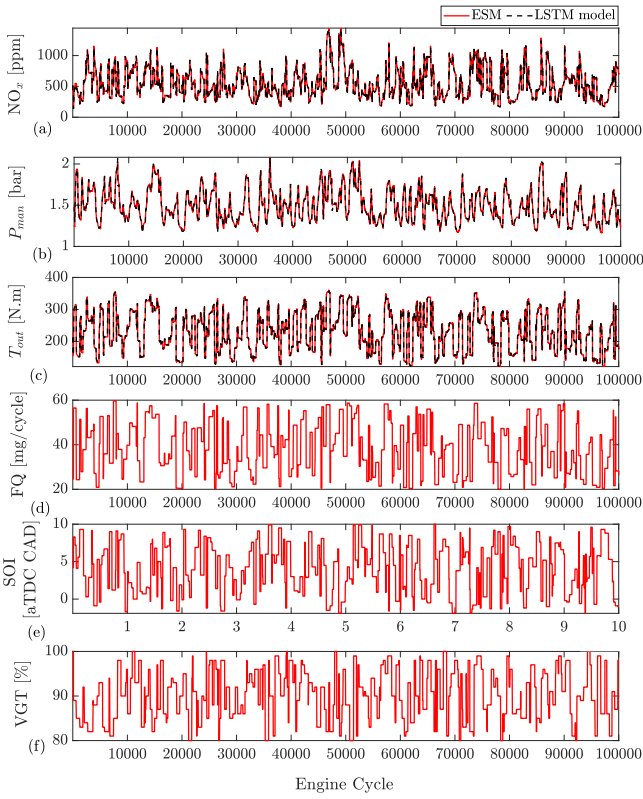
$$y(k) = [T_{out}(k) \ NO_x(k)]^T, \quad (17)$$

$$u(k) = [FQ(k) \ SOI(k) \ VGT(k)]^T.$$

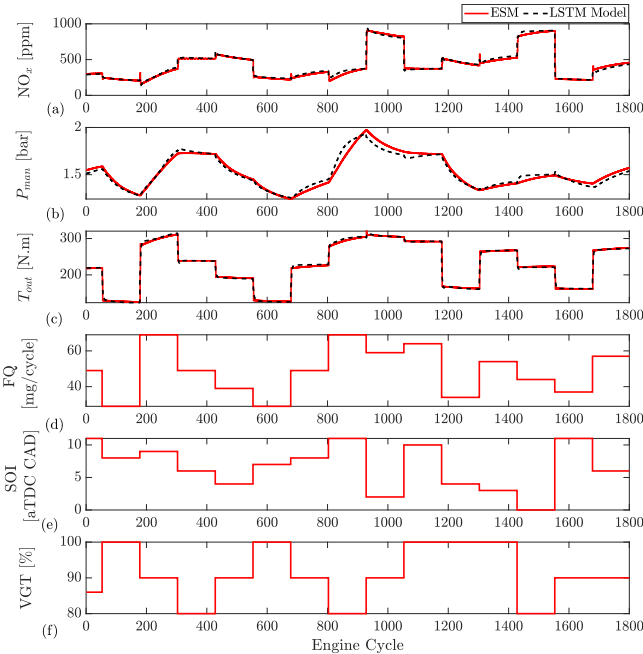
In the defined cost function (Eq. (14)) the error between output and requested torque are penalized along with NO<sub>x</sub> and fuel quantity minimization.

This leads to the definition of the following OCP, which is solved at each discrete-time instant:

$$\begin{aligned} \min_{u(\cdot|k), s(k)} \quad & J(u(\cdot|k), s(k)) \\ \text{s.t.} \quad & x(0) = \bar{x}(0) \\ & x(k+1) = f(x(k), u(k)) \quad k = 0, \dots, p-1 \\ & \underline{x} \leq x(k) \leq \bar{x} \quad k = 0, \dots, p \\ & \underline{u} \leq u(k) \leq \bar{u} \quad k = 0, \dots, p-1 \end{aligned} \quad (18)$$

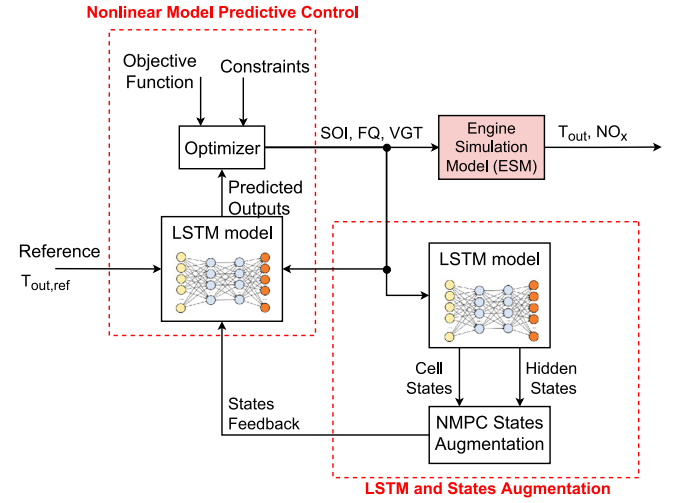


**Fig. 6.** Training and validation results for the LSTM model vs. ESM: (a) engine-out  $\text{NO}_x$ , (b) intake manifold pressure  $P_{\text{man}}$ , (c) engine-output torque  $T_{\text{out}}$ , (d) fuel quantity (FQ), (e) Start of injection (SOI), (f) Variable Geometry Turbine (VGT) rate—cycles 1 to 80,000 are devoted for training and cycles 80,001 to 100,000 used for validation.



**Fig. 7.** LSTM model comparison for engine-out emissions and performance: (a) engine-out  $\text{NO}_x$ , (b) intake manifold pressure ( $P_{\text{man}}$ ), (c) engine-output torque ( $T_{\text{out}}$ ), (d) fuel quantity (FQ), (e) Start of injection (SOI), (f) Variable Geometry Turbine (VGT) rate.

where  $x_{k+1} = f(x(k), u(k))$  is from the LSTM model where  $x$  represents the states of model. For the formulation of the LSTM-NMPC, the state vector of the MPC (Eq. (17)) has to be extended by adding the hidden



**Fig. 8.** Block diagram of LSTM-NMPC structure.

**Table 2**

Constraint values.

Min value ( $\bar{x}, \underline{u}$ )	Variable ( $x, u$ )	Max value ( $\bar{x}, \bar{u}$ )
0 ppm	$\text{NO}_x$	500 ppm
10 mg/cycle	FQ	80 mg/cycle
-2 aTDC CAD	SOI	11 aTDC CAD
70%	VGT	100%

and cell states of the LSTM network. Therefore, the prediction model of the LSTM-NMPC consists of the following updated state vector  $x(k)$ , the output vector  $y(k)$  and the control vector  $u(k)$  which are defined as

$$x = [h_{LSTM1}, h_{LSTM2}, c_{LSTM1}, c_{LSTM2}]^T,$$

$$y = [T_{\text{out}}, \text{NO}_x]^T,$$

$$u = [\text{FQ}, \text{SOI}, \text{VGT}]^T, .$$

The hidden and cell states result in a total of 104 states for the LSTM-MPC. Schematically the LSTM-NMPC is shown in Fig. 8. The nonlinear dynamics of the NMPC are defined by formulating the chain rule on the recurrent network's prediction and update functions, stated in Eqs. (1) and (4). The prediction horizon,  $p$ , has been chosen as 5 with a control horizon of one step.

The NMPC problem has 104 states, and all of these states must be estimated to update the NMPC block that is used in MATLAB/SIMULINK®. Therefore, the same dynamic model is used as an estimator for the hidden and cell states of the LSTM model to provide the 104 states of the NMPC problem.

The upper and lower constraints of the states and control output are given in Table 2. The upper limit on  $\text{NO}_x$  is used to limit the peak emissions values and this value can be adjusted to comply with emissions legislation. For this work, a limit of 500 ppm has been selected which is the maximum measured  $\text{NO}_x$  emissions at 1500 rpm when using the production ECU.

The limit imposed on FQ is not strictly required as the goal of the MPC is to minimize FQ. However, the limit is imposed to prevent the controller from attempting large fuel injection amounts. Small injection amounts outside of the injector calibration range are also restricted. There is both an upper and lower constraint applied to the SOI. Early SOI is limited to prevent extremely early combustion phasing which can lead to high engine noise and possible engine damage. Late SOI is also constrained to prevent low combustion efficiency and high exhaust gas temperatures. The constraints provided on VGT are used to match the defined turbine map which runs from 0.7–1.0.

The weights of LSTM-NMPC are manually tuned where the highest weight is given to output torque tracking and giving equal weights to

$\text{NO}_x$  and PM emissions and fuel consumption reduction. Then a weight equal to half that of the emissions is given to changes in control output. These tuned weights are  $w_{T_{\text{out}}} = 1$ ,  $w_{\text{NO}_x} = 0.2$ ,  $w_{FQ} = 0.2$ ,  $w_{\Delta u} = 0.1$ , respectively.

Here, the states, outputs, and control inputs are normalized using z-score normalization which is defined as

$$z_{\text{normal}} = \frac{z - \mu}{\sigma} \quad (19)$$

where  $\mu$  is the mean and  $\sigma$  is the standard deviation of  $z$  for both the input and output space. This allows for a reduction in the computational time for the normalization and denormalization required for evaluating the LSTM neural network in the NMPC problem. The constraint softening value was set as 0.1 which indicates hard constraints.

In this study, the MATLAB Toolbox utilizing the `fmincon` solver is used for the NMPC simulation. For comparison, state-of-the-art commercial solver FORCES PRO by EMBOTEC (Domahidi & Jerez, 2014–2019; Zanelli, Domahidi, Jerez, & Morari, 2017) and open-source package acados (Verschuere et al., 2018, 2021) with the QP solver HPIPM (High-Performance Interior-Point Method) (Frison & Diehl, 2020) are also used to solve the NMPC optimization. In NMPC, the OCP structured NLP is reformulated into a Sequential Quadratic Programming (SQP) problem by means of iterative quadratic approximations at the shooting nodes (Winkler et al., 2021). This results in sequential quadratic subproblems that can be solved and contribute to the holistic solution of the NLP by means of reformulations of the initial problem. The number of SQP iterations is called SQP steps which have a strong influence on the computational efficiency as well as the prediction quality of the NMPC. In the results and discussion section, the computational timing of `fmincon`, FORCES PRO, and acados with HPIPM are compared.

## 5. NMPC imitative controller

In addition to applying ML to modeling of the system it is also possible to utilize ML for control implementation. The imitation of NMPC using deep learning methods in so called imitative NMPC, is used to avoid the high computational time of online optimized NMPC (Zhang et al., 2019).

The application of imitative NMPC is shown schematically in Fig. 9 which shown the three main steps. In the first step, the previously designed LSTM-NMPC is implemented on the ESM. Second, the NMPC input and output are recorded, and a deep neural network, including a LSTM layer is used to fit the controller data to mimic the behavior of the NMPC. The final step in the process involves replacing the online NMPC with an imitative controller to avoid the high computational time of NMPC. That is, instead of solving NMPC optimization online, the identified function, here a deep network, is deployed with a much lower computation cost.

In order to clone the behavior of the NMPC, a deep network structure using LSTM is proposed. The proposed imitative LSTM-NMPC is shown schematically in Fig. 10. The engine-output torque  $T_{\text{out}}$ , the error in output torque ( $e_{T_{\text{out}}}$ ), engine-out  $\text{NO}_x$ , intake manifold pressure  $P_{\text{man}}$ , and engine speed  $n_{\text{rpm}}$  are the inputs of the imitative NMPC, and the goal here is to generate control action of fuel quantity (FQ), start of injection (SOI), and VGT by mimicking the previously designed NMPC controller.

Therefore, the forward propagation of this imitative network used to replace the online optimization of MPC is given as

$$z_{FC1}(k) = \text{ReLU}(W_{FC1}^T u(k) + b_{FC1}) \quad (20a)$$

$$h_{LSTM}(k) = \underbrace{o_{LSTM}(k) \odot \tanh(c_{LSTM}(k))}_{\text{based on Eq. (1)}} \quad (20b)$$

$$z_{FC2}(k) = \text{ReLU}(W_{FC2}^T h_{LSTM}(k) + b_{FC2}) \quad (20c)$$

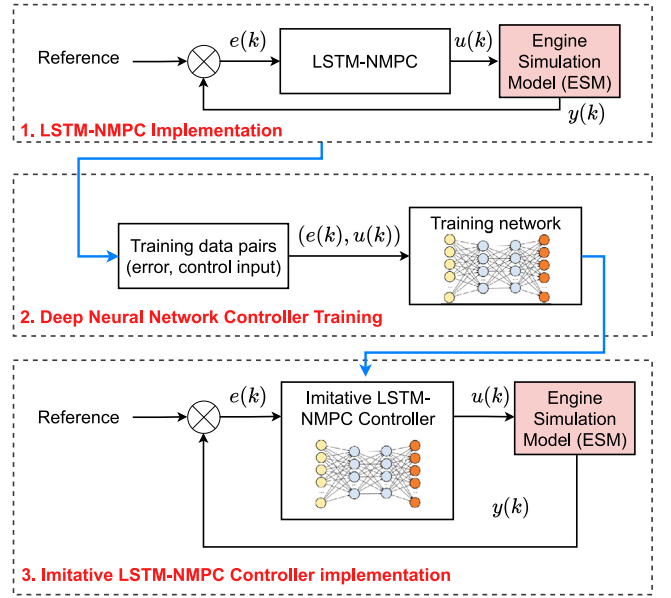


Fig. 9. Concept of Imitative NMPC – (1) Implementation of original LSTM-NMPC in ESM co-simulation, (2) Training deep neural network based on LSTM-NMPC collected data, (3) replace trained deep network (imitative controller) with LSTM-NMPC in ESM co-simulation – NMPC: Nonlinear Model Predictive Control, LSTM: Long-Short Term Memory, ESM: Engine Simulation Model.

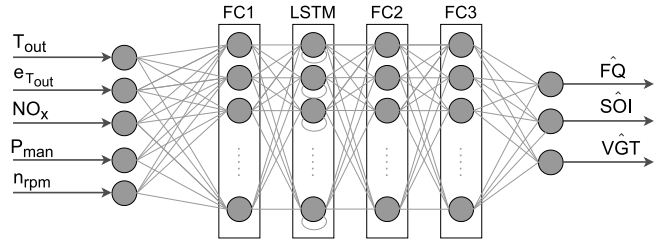


Fig. 10. Structure of proposed network for imitation of NMPC.

$$\underbrace{z_{FC3}(k)}_{\hat{u}(k)=[\hat{FQ}(k) \quad \hat{SOI}(k) \quad \hat{VGT}]^T} = \text{ReLU}(W_{FC3}^T z_{FC2}(k) + b_{FC3}) \quad (20d)$$

The output of the proposed imitative LSTM-NMPC network which is the estimated control actions can be calculated as

$$\hat{u}(k) = \begin{bmatrix} \hat{FQ}(k) \\ \hat{SOI}(k) \\ \hat{VGT} \end{bmatrix} \quad (21)$$

The cost function for this network is

$$J(W, b) = \frac{1}{m} \sum_{k=1}^m \mathcal{L}(\hat{u}(k), u(k)) + \frac{\lambda}{2m} \sum_{l=1}^L \|W^{[l]}\|_2^2 \quad (22)$$

where  $\mathcal{L}(\hat{u}(k), u(k))$  is the loss function,  $\lambda$  is the regularization coefficient, and  $\|W^{[l]}\|_2^2$  is the Euclidean norm which is defined as

$$\|W^{[l]}\|_2^2 = \sum_{i=1}^{n^{[l]}} \sum_{j=1}^{n^{[l-1]}} (w_{ij}^{[l]})^2 \quad (23)$$

The Mean Squared Error (MSE) cost function is used and is defined as

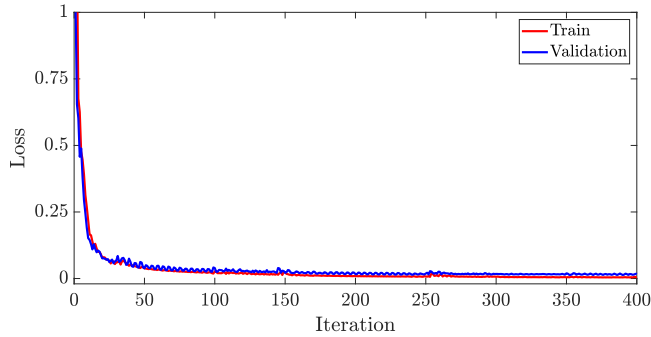
$$\mathcal{L}(\hat{u}(k), u(k)) = \frac{1}{m} \sum_{k=1}^m (\hat{u}(k) - u(k))^2 \quad (24)$$

The training details of the imitative controller is presented in Table 3. As the performance of the imitative controller depends on



**Table 3**  
Properties of imitative controller based on LSTM-NMPC.

Name	LSTM-NMPC
FC size	32
LSTM size	32
Optimizer	Adam
Maximum Epochs	400
Mini batch size	512
Learn rate drop period	150 Epochs
Learn rate drop factor	0.5
L2 Regularization	1
Initial learning rate	0.02
Validation frequency	1
Momentum	0.9
Squared gradient decay	0.99



**Fig. 11.** Imitative LSTM-NMPC loss function vs. iteration.

**Table 4**  
Imitative LSTM-NMPC controller train and validation accuracy.

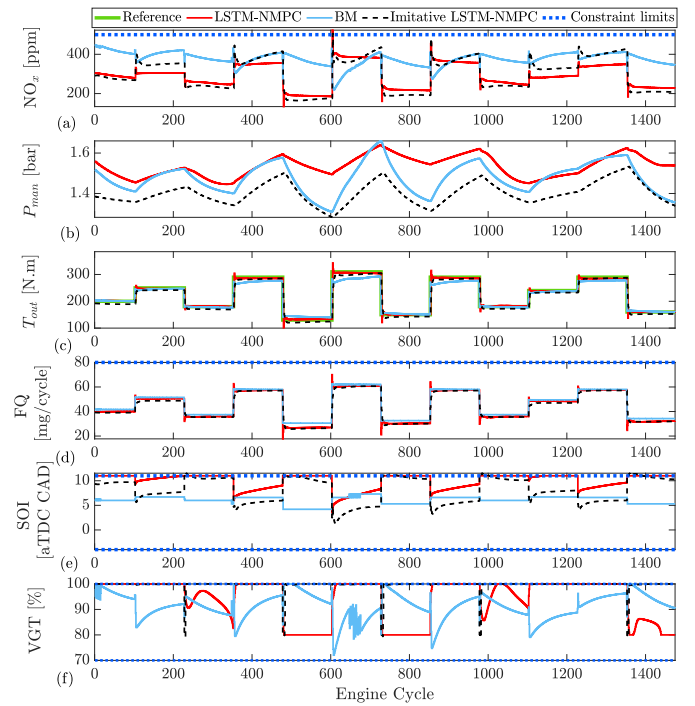
	FQ	SOI	VGT
Train	2.12%	4.55%	4.90%
Validation	2.47%	9.98%	12.46%

collected data, widely varying operating conditions need to be tested and fed into the network. To do this the NMPC controller was evaluated for randomly changing engine speeds from 1200 rpm to 1800 rpm and for a requested load ( $T_{out}$  reference) of 120–320 N m to make the imitative controller robust to a range of operating conditions. The loss function versus iteration is shown in Fig. 11. This loss function indicates the training process has been completed and has avoided the overfitting-underfitting problem as the validation data eventually converges to training loss.

To train the imitative controllers, the NMPC is evaluated for 2000 s of simulation, where 1600 s are devoted for training data and 400 s of the simulation are used as the validation data set. The RMSE of the training and validation set for are presented in Table 4. As shown, the DNN network can clone the NMPC behavior with average accuracy of 3.9% for training and 8.3% for validation. These imitative controllers are tested for newly generated references, and their performance against NMPC online optimization will be presented in the next section.

## 6. Results and discussions

The objective of the developed controller is to track a target load while minimizing  $NO_x$  and the fuel used in the diesel engine. Here the results of the NMPC controller with LSTM-based data driven model and imitative LSTM-NMPC controller are compared to the benchmark model (BM). The BM here is the calibrated ECU tables based on the Cummins production ECU that is embedded in GT-power for ESM co-simulation—more details are available in our previous studies (Shahpouri et al., 2021a, 2021b). Fig. 12 presents all the controller inputs and outputs for all three controllers tested.



**Fig. 12.** Controller comparison in engine speed of 1500 rpm: (a) engine-out  $NO_x$ , (b) intake manifold pressure ( $P_{man}$ ), (c) engine-output torque ( $T_{out}$ ), (d) fuel quantity (FQ), (e) Start of injection (SOI), (f) Variable Geometry Turbine (VGT) rate.

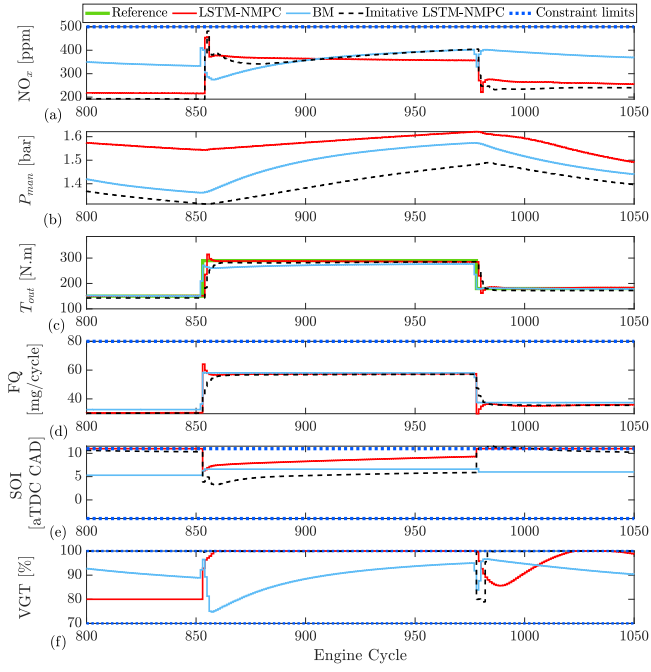
The tested controllers are subject to input constraints for  $NO_x$ , FQ, SOI and VGT. These limits are listed in Table 2. Slight constraint violations for  $NO_x$  at engine cycle 600 are seen. This is attributed to the constraint softening that is implemented in the NMPC. However, overall both the LSTM-NMPC and imitative controllers are able to keep the  $NO_x$  levels below the specified constraint. The BM is significantly worse than the nonlinear MPC model which is likely due to operating the engine not at one of the optimal calibration points in the feedforward tables.

The upper limit for SOI constraints is often flit. Here the controllers would like to implement later injection timing than is currently allowed. These late injection timings reduce the peak combustion temperature leading to lower  $NO_x$  levels, however, they also lead to reduced combustion efficiency and increased fuel consumption.

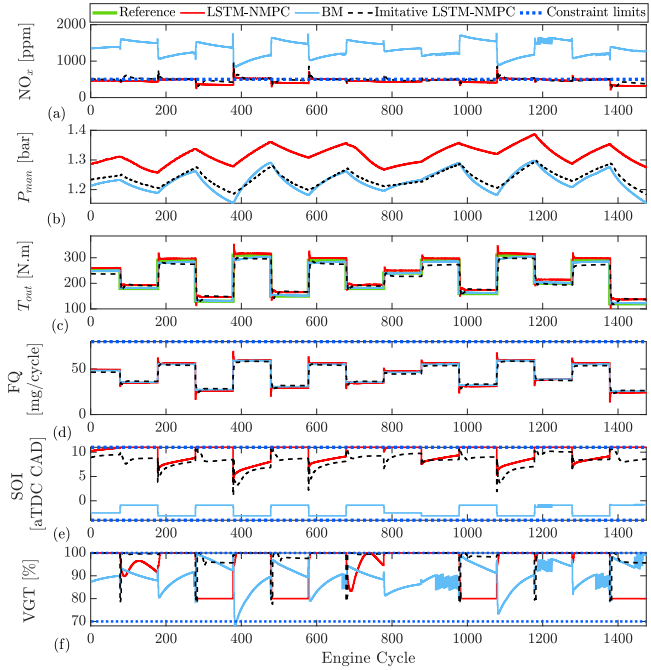
The torque ( $T_{out}$ ) tracking of both NMPC and BM are shown for a step in Fig. 13 and acceptable performance is achieved. Here the BM uses the most fuel and still is unable to reach the desired torque.

To check the controller's robustness, the engine speed is changed from 1500 rpm (where the NMPC model was designed and validated) to 1200 rpm. The NMPC performance at 1200 rpm can be seen in Fig. 14 (zoomed version from 450 to 650 is shown in Fig. 15). From these figures it can be seen that both the LSTM-NMPC and Imitative NMPC exhibit similar trends to their operation at 1500 rpm. Again the actions of the imitative controller are slower and without overshoot to the step inputs when compared to the online LSTM-NMPC. Both of these controllers outperform the BM production ECU in terms of  $NO_x$  production. As the speed is changed from 1500 rpm, the imitative controller has oscillations in the controller outputs as seen in fuel quantity, start of injection and VGT rate. The largest effect of changing speed is the significant increase in  $NO_x$  emissions of the BM. It is assumed that this is caused by deviating from 1500 rpm, the design operating point of the industrial engine. The high  $NO_x$  value at this speed in the BM is due to a significant advance in injection timing resulting in a significant increase in  $NO_x$  emissions.

To summarize the performance of the controllers at both engine speeds the cumulative and average  $NO_x$  emissions, load error, FQ and



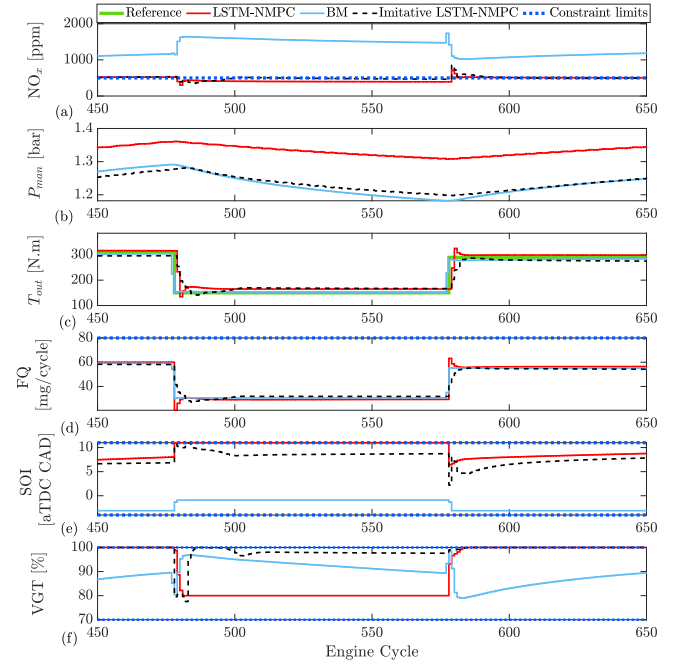
**Fig. 13.** Controller comparison in engine speed of 1500 rpm zoomed from 800 to 1050 cycles: (a) engine-out  $\text{NO}_x$ , (b) intake manifold pressure ( $P_{\text{man}}$ ), (c) engine-output torque ( $T_{\text{out}}$ ), (d) fuel quantity (FQ), (e) Start of injection (SOI), (f) Variable Geometry Turbine (VGT) rate.



**Fig. 14.** Controller comparison in engine speed of 1200 rpm: (a) engine-out  $\text{NO}_x$ , (b) intake manifold pressure ( $P_{\text{man}}$ ), (c) engine-output torque ( $T_{\text{out}}$ ), (d) fuel quantity (FQ), (e) Start of injection (SOI), (f) Variable Geometry Turbine (VGT) rate.

Execution time are shown in Table 6. For the design speed of 1500 rpm both the LSTM-NMPC and imitative NMPC outperform the BM in terms of  $\text{NO}_x$  emission with a slight reduction in fuel consumption.

To investigate the NMPC execution time, *acados* (QP solver HPIPM), *FORCES PRO*, and *fmincon* are each evaluated and their computational time are compared and summarized in Table 5. *acados* with HPIPM provides the fastest solve time among the solvers tested.



**Fig. 15.** Controller comparison in engine speed of 1200 rpm zoomed from 450 to 650 cycles: (a) engine-out  $\text{NO}_x$ , (b) intake manifold pressure ( $P_{\text{man}}$ ), (c) engine-output torque ( $T_{\text{out}}$ ), (d) fuel quantity (FQ), (e) Start of injection (SOI), (f) Variable Geometry Turbine (VGT) rate.

In the *acados* implementation, a maximum QP iteration of 50 and maximum NLP iteration/SQP steps of 5 are used, which results in an average runtime of 12.20 ms and maximum runtime of 31.56 ms for 1500 RPM. This value is much faster than the average runtimes of *fmincon* that required 786.02 ms. *FORCES PRO* was also tested and showed an improvement in runtime over the *fmincon* but was significantly slower than the *acados* implementation. The exact computational timing of *FORCES PRO* cannot be disclosed due to the academic license agreement.

The authors assume that this difference in the runtime can be attributed to the fully condensed problem formulation used in the *acados* implementation in comparison to the sparse formulation of *fmincon* and *FORCES PRO*. The presented OCP consists of a large state vector with 104 states and a relatively small prediction horizon with 5 steps and a relatively small control vector with 3 control outputs. This allows the condensed problem formulation used in *acados* to take full advantage of the condensation benefits (Diehl, Ferreau, & Haverbeke, 2009; Frison, Kouzoupis, Jørgensen, & Diehl, 2016). Another cause for the difference in runtime is attributed to the underlying algorithmic differentiation framework CasADi MX (Matrix expression) symbolic variables in *acados* instead of SX (scalar symbolic) variables in *FORCES PRO*. The general matrix expression type MX tends to be more computationally efficient when working with larger matrices (Andersson, Gillis, Horn, Rawlings, & Diehl, 2019).

Although the *acados* computational time of the online NMPC is orders of magnitude slower than imitation NMPC. The ML-based imitation controllers rely on MPC results, and thus an MPC must be designed and run to collect data for controller training. The ECU time of the benchmark model is estimated by recreating the production feedforward calibration tables on the MABX II. As there are various corrections applied to the calculated values, the BM controller actually takes longer to compute than the imitative controller. This is expected as the BM model considers various environmental factors such as engine and air temperatures as well as ambient pressures, which are not considered in the developed NMPC.

**Table 5**

Turnaround time comparison between Matlab `fmincon`, EMBOTECH FORCES PRO, and `acados` solvers—PIL: Processor in the loop.

Solver	Average turnaround time [ms]	Real-time verification in PIL setup
Matlab <code>fmincon</code>	786.02	x
EMBOTECH FORCES PRO	786.02 > t > 12.20	x
<code>acados</code>	12.20	✓

**Table 6**

Proposed MPC and Imitative MPC results compared to Benchmark for engine speed of 1500 and 1200 rpm.

1500 rpm						
	Cumulative NO <sub>x</sub> [ppm]	Average NO <sub>x</sub> [ppm]	Load error [%]	Cumulative FQ [g]	Average FQ [mg]	Execution time [ms] <sup>a</sup>
Benchmark	556 100.0	376.8	3.95	67.9	46.0	≈0.08 <sup>b</sup>
LSTM-NMPC	428 420.0	290.2	1.90	65.6	44.4	12.20 <sup>c</sup>
Imitative LSTM-NMPC	438 850.0	297.3	3.74	64.6	43.8	0.04
1200 rpm						
	Cumulative NO <sub>x</sub> [ppm]	Average NO <sub>x</sub> [ppm]	Load error [%]	Cumulative FQ [g]	Average FQ [mg]	Execution time [s] <sup>a</sup>
Benchmark	1 961 900.0	1329.2	2.18	64.8	43.9	≈0.08 <sup>b</sup>
LSTM-NMPC	671 650.0	455.0	6.95	64.9	43.9	11.50 <sup>b</sup>
Imitative LSTM-NMPC	718 380.0	486.7	7.61	64.0	43.4	0.03

<sup>a</sup>Per engine cycle of simulation.

<sup>b</sup>Exact production ECU time is unknown, estimated by recreating Cummins ECU on MABX II.

<sup>c</sup>`acados` execution time.

**Table 7**

Percentage of improvement for proposed MPC and Imitative MPC with respect to Benchmark for engine speed of 1500 and 1200 rpm.

1500 rpm			
	NO <sub>x</sub> [%]	FQ [%]	Load error [%]
LSTM-NMPC	−22.98	−3.48	+2.05
Imitative LSTM-NMPC	−21.10	−4.78	+0.21
1200 rpm			
	NO <sub>x</sub> [%]	FQ [%]	Load error [%]
LSTM-NMPC	−65.77	+0.15	−4.77
Imitative LSTM-NMPC	−63.38	−1.23	−5.43

The performance of the developed controllers compared to the BM can be seen in Table 7. Here significant NO<sub>x</sub> emissions reduction can be seen for the controller over the BM. In addition to these improvements in emissions the developed controllers use the same amount or less fuel than the baseline model. This demonstrates the advantage of the optimized controllers over the BM. The developed models do experience a slight increased load error in comparison to BM for 1500 rpm.

The developed controller in this study has also been compared to Linear Parameter Varying MPC (LPV-MPC) designed for ICE control in the literature (Irdmoussa et al., 2019). The LPV-MPC adjustment for the emission control of a CI engine was developed in our previous study (Norouzi et al., 2022) based on (Irdmoussa et al., 2019)—more details of this implementation are available in Norouzi et al. (2022). Table 8 shows the performance of LPV-MPC vs. LSTM-NMPC and imitation LSTM-NMPC. As a result of using a nonlinear optimization (LSTM-NMPC), the computational time is increased by 10.5 ms when compared to LPV-MPC. However, the imitation of LSTM-NMPC provides more than 40 times faster performance than the LPV-MPC. Both LSTM-NMPC and imitation LSTM-NMPC are capable of reducing NO<sub>x</sub> emission by 4.9% and 2.6% compared to LPV-NMPC. This NO<sub>x</sub> reduction is a trade-off with a slight increase in the fuel consumption of LSTM-NMPC.

Overall, the imitative NMPC controller provide similar improvements to the NMPC controller over the BM and LPV-MPC (Irdmoussa et al., 2019; Norouzi et al., 2022) but provides a significant decrease in

computational time. This makes future real-time implementation while minimizing the computational power needed with a full MPC feasible.

## 7. Conclusions

The integration of machine learning and model predictive control for both modeling and controller implementation is discussed. First, using a deep recurrent neural network, a long–short term memory (LSTM) network is designed to predict diesel engine performance and emissions. A nonlinear model predictive controller (NMPC) is designed based on this network and by augmenting hidden and cell states. The prediction accuracy of these models for test data shows acceptable results. This model accuracy is expected as the LSTM is capable of a more generalizable prediction since it uses hidden and cell states to capture long term relationships within the data.

In addition to using ML for modeling ML can also be implemented as a controller where an NMPC is implemented and data is collected from the MPC input and output which is used to train a deep neural network. By replacing the full online MPC by an ML based imitation it is possible to reduce the computational time of an NMPC. This imitative controller is compared with a baseline Cummins calibrated ECU model using the Engine Simulation Model (ESM) in a GT-power/MATLAB/SIMULINK co-simulation.

Minimizing NO<sub>x</sub> emissions and reducing the injected fuel amount while maintaining the same load is the main goal of the controller. The developed controllers are also constrained to meet NO<sub>x</sub> legislated limits in addition to constraints on all inputs to ensure engine safety. The developed controllers are simulated using the ESM co-simulation. To evaluate the robustness of the controllers, the engine speed is changed from 1500 rpm where the NMPC model were validated to 1200 rpm. All of the controllers produce significant NO<sub>x</sub> reduction, especially at lower engine speeds with respect to benchmark feedforward controller. The NO<sub>x</sub> reduction for 1500 and 1200 rpm for NMPC are 23.0% and 65.8%. The imitative controller successfully cloned the NMPC behavior with a NO<sub>x</sub> reduction of 21.1% at 1500 rpm. At 1200 rpm the reduction is 63.4% when compared to the BM. The imitative controller performs similarly to the online MPC by learning from the MPC experiment but requires a much lower computational time. The computation time for

Table 8

Proposed MPC and Imitative MPC results compared to Linear Parameter Varying MPC (LPV-MPC) (Irdmoussa et al., 2019; Norouzi et al., 2022) for engine speeds of 1500.

	Cumulative NOx [ppm]	Average NOx [ppm]	Load error [%]	Cumulative FQ [g]	Average FQ [mg]	Execution time [ms]
LPV-MPC (Irdmoussa et al., 2019; Norouzi et al., 2022)	450570.0	305.3	2.96	65.5	44.4	1.69
LSTM-NMPC	428420.0	290.2	1.90	65.6	44.4	12.20
Imitative LSTM-NMPC	438850.0	297.3	3.74	64.6	43.8	0.04

the imitative controller is a factor of 100 lower than the online optimized MPC. Real-time implementation of LSTM-NMPC and imitative LSTM-NMPC controllers on the engine is planned as future work.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgments

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The research was performed as part of the Research Group (Forschungsgruppe) FOR 2401 “Optimization based Multiscale Control for Low Temperature Combustion Engines”, which is funded by the German Research Association (Deutsche Forschungsgemeinschaft, DFG) and with Natural Sciences Research Council of Canada Grant 2016–04646. Partial funding from Future Energy Systems and Alberta innovates at the University of Alberta, Canada is also gratefully acknowledged.

### References

- Andersson, Joel A E, Gillis, Joris, Horn, Greg, Rawlings, James B, & Diehl, Moritz (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36.
- Asprion, Jonas, Chinellato, Oscar, & Guzzella, Lino (2014). Optimal control of diesel engines: numerical methods, applications, and experimental validation. *Mathematical Problems in Engineering*, Article ID 286538, 1–21.
- Basina, L. N., Aditya, Irdmoussa, Behrouz K., Velni, Javad Mohammadpour, Borhan, Hoseinali, Naber, Jeffrey D., & Shahbakhti, Mahdi (2020). Data-driven modeling and predictive control of maximum pressure rise rate in RCCI engines. In *IEEE conference on control technology and applications (CCTA 2020)* (pp. 94–99). IEEE.
- Bemporad, Alberto (2020). Model predictive control, quadratic programming and explicit MPC. In *Model Predictive Control*. IMT Lucca.
- Bemporad, Alberto, Bernardini, Daniele, Long, Ruixing, & Verdejo, Julian (2018). Model predictive control of turbocharged gasoline engines for mass production. In *WCX world congress experience*. SAE International.
- Bromnick, Paul (1999). *Development of a model predictive controller for engine idle speed using cpower*. SAE paper no. 1999-01-1171, SAE International.
- Broomhead, Timothy, Manzie, Chris, Hield, Peter, Shekhar, Rohan, & Brear, Michael (2016). Economic model predictive control and applications for diesel generators. *IEEE Transactions on Control Systems Technology*, 25(2), 388–400.
- Cera, Brian, & Agogino, Alice M. (2018). Multi-cable rolling locomotion with spherical tensegrities using model predictive control and deep learning. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1–9). IEEE.
- Dahl, J., Wassén, H., Santin, O., Herceg, M., Lansky, L., Pekar, J., et al. (2018). Model predictive control of a diesel engine with turbo compound and exhaust after-treatment constraints. *IFAC-PapersOnLine*, 51(31), 349–354, 5th IFAC Conference on Engine and Powertrain Control, Simulation and Modeling E-COSM 2018.
- Deb, Kalyanmoy, & Jain, Himanshu (2013). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4), 577–601.
- Diehl, Moritz, Ferreau, Hans Joachim, & Haverbeke, Niels (2009). Efficient numerical methods for nonlinear MPC and moving horizon estimation. In Lalo Magni, Davide Martino Raimondo, & Frank Allgöwer (Eds.), *Nonlinear model predictive control: towards new challenging applications* (pp. 391–417). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Domahidi, Alexander, & Jerez, Juan (2014–2019). FORCES professional. Embotech AG, <https://embotech.com/FORCES-Pro>.
- Drgoňa, Ján, Picard, Damien, Kvasnica, Michal, & Helsen, Lieve (2018). Approximate model predictive building control via machine learning. *Applied Energy*, 218, 199–216.
- Frison, Gianluca, & Diehl, Moritz (2020). HPIPM: a high-performance quadratic programming framework for model predictive control. *IFAC-PapersOnLine*, 53(2), 6563–6569, 21th IFAC World Congress.
- Frison, Gianluca, Kouzoupis, Dimitris, Jørgensen, John, & Diehl, Moritz (2016). An efficient implementation of partial condensing for nonlinear model predictive control. In *2016 IEEE 55th conference on decision and control (CDC)* (pp. 4457–4462).
- Géron, Aurélien (2019). *Hands-on machine learning with scikit-learn, keras, and tensorflow: concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.
- Gordon, David, Norouzi, Armin, Blomeyer, Gero, Bedei, Julian, Aliramezani, Masoud, Andert, Jakob, et al. 2021. Support vector machine based emissions modeling using particle swarm optimization for homogeneous charge compression ignition engine. *International Journal of Engine Research*. 14680874211055546.
- Guzzella, Lino, & Onder, Christopher (2009). *Introduction to modeling and control of internal combustion engine systems*. Springer Science & Business Media.
- Herceg, Martin, Raff, Tobias, Findeisen, Rolf, & Allgöwer, Frank (2006). Nonlinear model predictive control of a turbocharged diesel engine. In *2006 IEEE conference on computer aided control system design, 2006 IEEE international conference on control applications, 2006 IEEE international symposium on intelligent control* (pp. 2766–2771).
- Hewing, Lukas, Wabersich, Kim P., Menner, Marcel, & Zeilinger, Melanie N. (2020). Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3, 269–296.
- Hu, Yunfeng, Chen, Huan, Wang, Ping, Chen, Hong, & Ren, Luquan (2018). Nonlinear model predictive controller design based on learning model for turbocharged gasoline engine of passenger vehicle. *Mechanical Systems and Signal Processing*, 109, 74–88.
- Huang, Mike, Liao-McPherson, Dominic, Kim, Shinhoon, Butts, Ken, & Kolmanovsky, Ilya (2018). Toward real-time automotive model predictive control: A perspective from a diesel air path control development. In *American control conference (ACC 2018)* (pp. 2425–2430). IEEE.
- Irdmoussa, B., Khoshbakht, Rizvi, S. Z., Velni, J., Mohammadpour, Naber, J., & Shahbakhti, M. (2019). Data-driven modeling and predictive control of combustion phasing for RCCI engines. In *American control conference (ACC 2019)* (pp. 1–6).
- Isermann, Rolf (2014). *Engine modeling and control*, Vol. 1017. Berlin: Springer Berlin Heidelberg.
- Janakiraman, Vijay Manikandan, Nguyen, XuanLong, & Assanis, Dennis (2016). An ELM based predictive control method for HCCI engines. *Engineering Applications of Artificial Intelligence*, 48, 106–118.
- Karg, Benjamin, & Lucia, Sergio (2018). Deep learning-based embedded mixed-integer model predictive control. In *2018 European control conference (ECC)* (pp. 2075–2080). IEEE.
- Karlsson, Maria, Ekholm, Kent, Strandh, Petter, Johansson, Rolf, & Tunestål, Per (2010). Multiple-input multiple-output model predictive control of a diesel engine. *IFAC Proceedings Volumes*, 43(7), 131–136, 6th IFAC Symposium on Advances in Automotive Control.
- Khoshbakht Irdmoussa, B., Naber, J., Mohammadpour Velni, J., Borhan, H., & Shahbakhti, M. (2021). Input-output data-driven modeling and MIMO predictive control of an RCCI engine combustion. *IFAC-PapersOnLine*, Modeling, Estimation and Control Conference (MECC 2021).
- Lennox, B., Montague, G. A., Frith, A. M., & Beaumont, A. J. (1998). Non-linear model-based predictive control of gasoline engine air-fuel ratio. *Transactions of the Institute of Measurement and Control*, 20(2), 103–112.
- Liao-McPherson, Dominic, Huang, Mike, Kim, Shinhoon, Shimada, Masanori, Butts, Ken, & Kolmanovsky, Ilya (2020). Model predictive emissions control of a diesel engine airpath: Design and experimental evaluation. *International Journal of Robust Nonlinear Control*, 30(17), 7446–7477.
- Liu, Zihao, Dizqah, Arash M., Herreros, Jose M., Schaub, Joschka, & Haas, Olivier (2021). Simultaneous control of NOx, soot and fuel economy of a diesel engine with dual-loop EGR and VNT using economic MPC. *Control Engineering Practice*, 108, Article 104701.
- López, José David, Espinosa, Jairo José, & Agudelo, John Ramiro (2011). LQR control for speed and torque of internal combustion engines. *IFAC Proceedings Volumes*, 44(1), 2230–2235, 18th IFAC World Congress.
- Moriyasu, Ryuta, Nojiri, Sayaka, Matsunaga, Akio, Nakamura, Toshihiro, & Jimbo, Tomohiko (2019). Diesel engine air path control based on neural approximation of nonlinear MPC. *Control Engineering Practice*, 91, Article 104114.
- Moriyasu, R., Ueda, M., Ikeda, T., Nagaoka, M., Jimbo, T., Matsunaga, A., et al. (2018). Real-time MPC design based on machine learning for a diesel engine air path system. *IFAC-PapersOnLine*, 51(31), 542–548, 5th IFAC Conference on Engine and Powertrain Control, Simulation and Modeling E-COSM 2018.



- Norouzi, Armin, Aliramezani, Masoud, & Koch, Charles Robert (2021). A correlation-based model order reduction approach for a diesel engine NOx and brake mean effective pressure dynamic model using machine learning. *International Journal of Engine Research*, 22(8), 2654–2672.
- Norouzi, Armin, Ebrahimi, Khashayar, & Koch, Charles Robert (2019). Integral discrete-time sliding mode control of homogeneous charge compression ignition (HCCI) engine load and combustion timing. *IFAC-PapersOnLine*, 52(5), 153–158.
- Norouzi, Armin, Gordon, David, Aliramezani, Masoud, & Koch, Charles Robert (2020). Machine learning-based diesel engine-out NOx reduction using a plug-in PD-type iterative learning control. In *2020 IEEE Conference on Control Technology and Applications (CCTA)* (pp. 450–455). IEEE.
- Norouzi, Armin, Heidarifard, Hamed, Shahbakhti, Mahdi, Koch, Charles Robert, & Borhan, Hoseinali (2021). Model predictive control of internal combustion engines: A review and future directions. *Energies*, 14(19).
- Norouzi, Armin, Shahpour, Saeid, Gordon, David, Winkler, Alexander, Nuss, Eugen, Abel, Drik, et al. (2022). Machine learning integrated with model predictive control for imitative optimal control of compression ignition engines. arXiv preprint arXiv:2204.00142.
- Novak, Mateja, & Dragicevic, Tomislav (2020). Supervised imitation learning of finite set model predictive control systems for power electronics. *IEEE Transactions on Industrial Electronics*.
- Nuss, Eugen, Wick, Maximilian, Andert, Jakob, De Schutter, Jochem, Diehl, Moritz, Abel, Dirk, et al. (2019). Nonlinear model predictive control of a discrete-cycle gasoline-controlled auto ignition engine model: Simulative analysis. *International Journal of Engine Research*, 20(10), 1025–1036.
- Peng, Qian, Huo, Da, & Hall, Carrie M. (2022). Neural network-based air handling control for modern diesel engines. *Proceedings of the Institution of Mechanical Engineers, Part D (Journal of Automobile Engineering)*, Article 09544070221083367.
- Pfeiffer, Roland, Haraldsson, Göran, Olsson, J-O, Tunestål, Per, Johansson, Rolf, & Johansson, Bengt (2004). System identification and LQG control of variable-compression HCCI engine dynamics. In *Proceedings of the 2004 IEEE international conference on control applications, 2004*, Vol. 2 (pp. 1442–1447). IEEE.
- Powell, J. D. (1987). A review of IC engine models for control system design. *IFAC Proceedings Volumes*, 20(5, Part 3), 235–240, 10th Triennial IFAC Congress on Automatic Control - 1987 Volume III, Munich, Germany, 27–31 July.
- Saerens, Bart, Diehl, Moritz, Swevers, Jan, & Van den Bulck, Eric (2008). Model predictive control of automotive powertrains-first experimental results. In *2008 47th IEEE conference on decision and control* (pp. 5692–5697). IEEE.
- Shahpour, Saeid, Norouzi, Armin, Hayduk, Christopher, Rezaei, Reza, Shahbakhti, Mahdi, & Koch, Charles Robert (2021a). Hybrid machine learning approaches and a systematic model selection process for predicting soot emissions in compression ignition engines. *Energies*, 14(23).
- Shahpour, Saeid, Norouzi, Armin, Hayduk, Christopher, Rezaei, Reza, Shahbakhti, Mahdi, & Koch, Charles Robert (2021b). Soot emission modeling of a compression ignition engine using machine learning. *IFAC-PapersOnLine*, Modeling, Estimation and Control Conference (MECC 2021).
- Souder, Jason S., & Hedrick, J. Karl (2004). Adaptive sliding mode control of air-fuel ratio in internal combustion engines. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 14(6), 525–541.
- Stewart, Gregory, & Borrelli, Francesco (2008). A model predictive control framework for industrial turbodiesel engine control. In *2008 47th IEEE conference on decision and control* (pp. 5704–5711). IEEE.
- Sun, Liting, Peng, Cheng, Zhan, Wei, & Tomizuka, Masayoshi (2018). A fast integrated planning and control framework for autonomous driving via imitation learning. In *Dynamic systems and control conference*, Vol. 51913. American Society of Mechanical Engineers, Article V003T37A012.
- Vaughan, A., & Bohac, S. V. (2015). Real-time, adaptive machine learning for non-stationary, near chaotic gasoline engine combustion time series. *Neural Networks*, 70, 18–26.
- Verschueren, Robin, Frison, Gianluca, Kouzoupis, Dimitris, van Duijkeren, Niels, Zanelli, Andrea, Quirynen, Rien, et al. 2018. Towards a modular software package for embedded optimization. In *Proceedings of the IFAC conference on nonlinear model predictive control (NMPC)*.
- Verschueren, Robin, Frison, Gianluca, Kouzoupis, Dimitris, Frey, Jonathan, van Duijkeren, Niels, Zanelli, Andrea, et al. (2021). acados – a modular open-source framework for fast embedded optimal control. *Mathematical Programming Computation*.
- Winkler, Alexander, Frey, Jonathan, Fahrbach, Timm, Frison, Gianluca, Scheer, René, Diehl, Moritz, et al. (2021). Embedded real-time nonlinear model predictive control for the thermal torque derating of an electric vehicle. *IFAC-PapersOnLine*, 54(6), 359–364, 7th IFAC Conference on Nonlinear Model Predictive Control NMPC 2021.
- Yashiro, Y., Jiwariyavej, V., Yamashita, Y., Hoshi, T., Terakado, K., & Ibaraki, S. (2017). High-speed model predictive control for next-generation turbocharging system. *Mitsubishi Heavy Industries Technical Review*, 54(1).
- Yin, Lianhao, Turesson, Gabriel, Tunestål, Per, & Johansson, Rolf (2020). Model predictive control of an advanced multiple cylinder engine with partially premixed combustion concept. *IEEE/ASME Transactions on Mechatronics*, 25(2), 804–814.
- Zanelli, A., Domahidi, A., Jerez, J., & Morari, M. (2017). FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs. *International Journal of Control*, 1–17.
- Zhang, Xiaojing, Bujarbaruah, Monimoy, & Borrelli, Francesco (2019). Safe and near-optimal policy learning for model predictive control using primal-dual neural networks. In *2019 American control conference (ACC)* (pp. 354–359). IEEE.
- Zhao, Dezong, Liu, Cunjia, Stobart, Richard, Deng, Jiamei, Winward, Edward, & Dong, Guangyu (2014). An explicit model predictive control framework for turbocharged diesel engines. *IEEE Transactions on Industrial Electronics*, 61(7), 3540–3552.
- Zhu, Qilun, Prucka, Robert, Prucka, Michael, & Dourra, Hussein (2017). A nonlinear model predictive control strategy with a disturbance observer for spark ignition engines with external EGR. *SAE International Journal of Commercial Vehicles*, 10(1), 360–372.