Control Engineering Practice 150 (2024) 106005

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Uncertainty-aware output feedback model predictive combustion control of RCCI engines☆

Pegah GhafGhanbari [a], Yajie Bao [b], Javad Mohammadpour Velni [a,*]

[a] Department of Mechanical Engineering, Clemson University, Clemson, SC, 29634, USA
[b] Intelligent Fusion Technology, Inc., Germantown, MD, 20874, USA

ARTICLE INFO

Keywords:
Scenario-based model predictive control
Inputs- and outputs-dependent uncertainty quantification
Reactivity controlled compression ignition engines
Linear parameter-varying framework

ABSTRACT

Accurate model development is essential for effective model-based control of Reactivity Controlled Compression Ignition (RCCI) engines. However, due to the intricate nature of engine combustion process, achieving a precise model that can capture the complex dynamic behavior and ensure high control performance poses a significant challenge. In this paper, we propose an uncertainty-aware output feedback model predictive control approach for efficient combustion management in RCCI engines. In contrast to the previously developed approaches, this method adopts a data-driven approach within the linear parameter-varying (LPV) framework for model development. To address the model mismatch between the LPV model and the real system/data, Bayesian Neural Networks (BNNs) are employed which provide the probability distribution of the uncertainties. The BNNs enable the formation of a scenario tree, effectively characterizing the range of potential uncertainties in the system. Through the implementation of scenario-based model predictive control, our approach ensures high tracking performance for the RCCI engine in the presence of modeling uncertainties and measurement noise. Extensive simulations and experimental validations demonstrate the superiority of our uncertainty-aware model predictive control over traditional control strategies.

1. Introduction

Internal combustion engines (ICE) have become the predominant power source in various applications, especially in transportation, yet controlling their emissions, a significant environmental concern, remains a challenge. Despite the proposal of alternative solutions such as hydrogen fuel cells and battery electric vehicles, widespread adoption faces obstacles, leaving ICE as the prevailing power source. In response to environmental challenges, low-temperature combustion (LTC) engines have been proposed, with Reactivity Controlled Compression Ignition (RCCI) offering promising characteristics (Lu, Han, & Huang, 2011). RCCI combines the use of low-reactivity fuels injected through the port fuel injectors (PFI) and high-reactivity fuels directly injected into the cylinder, achieving controlled and efficient combustion. The unique fuel combination and stratification enable RCCI to attain higher thermal efficiency and a broader operating range compared to conventional ICE engines, making it a potential solution for reducing emissions while also enhancing engine performance (Agarwal, Singh, & Maurya, 2017; Krishnamoorthi, Malayalamurthi, He, & Kandasamy, 2019; Singh, Kumar, & Agarwal, 2020).

Managing the combustion process in RCCI engines poses significant challenges due to their unique nature of ignition, which starts in high reactivity region and propagates to low reactivity region within the combustion chamber. Raut, Bidarvatan, Borhan and Shahbakhti (2018) developed a model-based control approach, including physics-based dynamic modeling and switched model predictive control (MPC), to achieve precise cycle-to-cycle control of combustion phasing ($CA50$) and indicated mean effective pressure ($IMEP$). Batool, Naber, and Shahbakhti (2021) focused on stable engine operation by controlling combustion cyclic variability, characterized by the coefficient of variation of IMEP (COV_{IMEP}). They integrated a data-driven model of COV_{IMEP} with physics-based models of $CA50$ and $IMEP$. Then, nonlinear MPC was applied to regulate $CA50$ and $IMEP$ while constraining COV_{IMEP}. Sitaraman et al. (2022) utilized a data-driven LPV model of an RCCI engine with scheduling variables defined based on early and late heat release rate fractions. They employed MPC to control CA50 and IMEP while also limiting Mean Pressure Rise Rate ($MPRR$) to prevent engine knocking.

The MPC schemes employed in the aforementioned studies provide limited robustness to (even) small uncertainties by re-initializing

☆ This work was supported by the US National Science Foundation under award CMMI-2302217.
* Corresponding author.
E-mail addresses: pghafgh@clemson.edu (P. GhafGhanbari), yajie.bao@intfusiontech.com (Y. Bao), javadm@clemson.edu (J. Mohammadpour Velni).

https://doi.org/10.1016/j.conengprac.2024.106005
Received 5 December 2023; Received in revised form 7 April 2024; Accepted 7 June 2024
Available online 29 June 2024

the optimizer at each sampling time. However, this approach is not sufficient in scenarios with significant uncertainties or model mismatch, as it does not guarantee closed-loop stability or constraint satisfaction and may result in poor tracking performance. To address this challenge, researchers have explored robustification methods for MPC, including min–max MPC (Campo & Morari, 1987), stochastic MPC (Mesbah, 2016), and tube-based MPC (Mayne, Seron, & Raković, 2005). Scenario-based MPC (ScMPC) (Bernardini & Bemporad, 2009) can strike a balance between robustness and computational efficiency while addressing over-conservatism. It copes with uncertainties by incorporating multiple possible scenarios or realizations, forming a scenario tree. Each scenario enables ScMPC to compute distinct control trajectories, allowing the controller to anticipate various system behaviors and adapt its control actions dynamically. The inclusion of feedback or recourse further enhances ScMPC's adaptability by adjusting decisions based on real-time feedback.

While ScMPC offers improved control performance and stability, careful attention is needed when selecting and representing scenarios to strike a balance between computational complexity and control effectiveness. Various techniques have been suggested to address this concern, with data-driven methods emerging as a particularly effective tool. In Bonzanini, Paulson, Makrygiorgos, and Mesbah (2021), Gaussian Processes (GPs) were employed to learn the state- and input-dependent plant-model mismatch in real-time. In contrast, Bao, Chan, Mesbah, and Velni (2023) used Bayesian Neural Networks (BNN) to quantify this mismatch, resulting in reduced conservativeness and improved control performance over GPs. However, it is important to note that these approaches assume the availability of all state information online, which may not be the case in practice. To address this limitation, input- and output-dependent uncertainties are quantified (learned) in this work using the BNN method, which provides information about the posterior distribution of the uncertainties. Compared with GPs, BNNs offer a more computationally efficient approach for uncertainty estimation, particularly for larger datasets, as BNNs leverage neural networks and optimization techniques, while GP can be computationally expensive due to matrix inversions and cubic scaling with the data size.

The main contribution of this work lies in the ***development and application of a scenario-based output feedback model predictive control scheme tailored for RCCI engines operating under uncertainties and disturbances***. The intricate and hard-to-model dynamics of RCCI engines pose considerable complexities, further compounded by limited output measurements. To address these challenges, we initially construct a data-driven model of the RCCI engine within the LPV framework, offering good accuracy but raising concerns about inherent errors in the learning process. This is mitigated by ScMPC which enhances the system's robustness against plant-model mismatch. Given the limited observability, we quantify this mismatch offline using a BNN with input and output information. It is shown that this approach effectively predicts modeling uncertainties, facilitating the generation of scenarios online for the proposed ScMPC. The probabilities associated with these scenarios are estimated using the moment matching method. Additionally, the limited availability of measurements poses challenges in control design. Here, the system states are estimated using a Kalman filter, known for its robustness to noisy measurements and provision of accurate state estimates crucial for effective control. The infeasibility problem that is highly probable to occur in the ScMPC framework is avoided through the implementation of soft constraints. Overall, *the proposed work contributes to the advancement of RCCI engine control by effectively managing uncertainties and improving the closed-loop system performance*.

The structure of this paper is as follows. Initially, a control-oriented model for the RCCI engine is developed, followed by an exposition of the state estimation technique utilizing the Kalman filter. Subsequently, the scenario-based model predictive control approach is introduced, providing an intricate elucidation of both scenario generation and probability calculation methods. The ensuing section offers the presentation of simulation results. The paper concludes with a concise summary encapsulating the key findings.

2. Control-oriented modeling of RCCI engine

Generally, state-space representation of a discrete-time nonlinear system is as follows

$$x_{k+1} = f(x_k, u_k) + w_k, \tag{1a}$$

$$y_k = h(x_k, u_k) + v_k, \tag{1b}$$

where x_k, u_k, and y_k are the state, input, and output vectors at time instant $k \in \mathbb{N}$. Moreover, $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \mapsto \mathbb{R}^{n_x}$ and $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \mapsto \mathbb{R}^{n_y}$ represent the dynamic and measurement models, and $w_k \sim \mathcal{N}(0, \mathcal{Q})$ and $v_k \sim \mathcal{N}(0, \mathcal{R})$ denote the normally distributed process and measurement noises, respectively.

The nonlinear dynamic model, $f(x_k, u_k)$, can be approximated with an LPV model, opening up opportunities to leverage the benefits of a linear control framework, while achieving improved model accuracy across a diverse range of operating conditions. Hence, the dynamic model (1) in the LPV framework is described as

$$x_{k+1} = A(\mathrm{p}_k)x_k + B(\mathrm{p}_k)u_k + \tilde{w}_k, \tag{2a}$$

$$y_k = C(\mathrm{p}_k)x_k + v_k, \tag{2b}$$

where $A : \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_x \times n_x}$, $B : \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_x \times n_u}$, and $C : \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_y \times n_x}$ denote state, input and output matrices, which are functions of time-varying scheduling variables $\mathrm{p}_k \in \mathbb{R}^{n_p}$. Furthermore, when describing the nonlinear model in the LPV framework, there arises a modeling mismatch, which, together with the process noise, are represented as a lumped term, $\tilde{w}_k$.

Control of RCCI engines, unlike their counterparts, requires adjusting numerous parameters, and this needs to be reflected in control-oriented model development. The average pressure exerted on the piston during the power stroke, indicated by IMEP, serves as a key metric for evaluating engine power and efficiency. Elevated maximum pressure rise rate (MPRR) can potentially result in engine knocking, a phenomenon that must be minimized. The timing at which 50% of the heat release occurs during the combustion process, known as CA50, plays a critical role in optimizing power delivery and fuel efficiency. Extensive research indicates that by incorporating key parameters such as start of injection (SOI) timing, premixed ratio (PR), and fuel quantity (FQ), as well as MPRR, IMEP, and CA50, control-oriented models can accurately capture the intricate dynamics of RCCI combustion (Basina et al., 2020; Irdmousa, Rizvi, Velni, Nabert, & Shahbakhti, 2019). Thus, the following input and output vectors are considered for state-space modeling of the RCCI engine (Sitaraman et al., 2022):

$$u = \begin{bmatrix} SOI & FQ & PR \end{bmatrix}^T, \tag{3a}$$

$$y = \begin{bmatrix} CA50 & MPRR & IMEP \end{bmatrix}^T. \tag{3b}$$

The heat release rate is another critical factor in combustion engines, significantly affecting their performance, efficiency, and emissions. This rate signifies the speed at which energy is released during combustion, reflecting the conversion rate from fuel-air mixture to thermal energy. It includes two primary aspects: the timing and magnitude of heat release. Early and rapid heat release can lead to abrupt pressure rises, while delayed release may result in incomplete combustion. Its magnitude and shape significantly influence key combustion characteristics such as peak pressure, combustion duration, brake thermal efficiency, and overall stability. Consequently, the fractions of early and late heat release, denoted as p_e and p_l, respectively, are utilized as scheduling variables in our LPV model (Sitaraman et al., 2022). This study employs a model structure similar to that presented by Sitaraman et al. (2022), featuring five states and a learned LPV model using the regularized least-squares support vector machine (LS-SVM) technique. However, the lack of knowledge regarding the system's internal states presents challenges that will be addressed in subsequent sections.

3. Parameter-dependent state estimation

In various practical applications, including the combustion control in RCCI engines, not all states of the system are directly measurable. Instead, only noisy measurements from certain sensors are accessible. However, since MPC relies on complete state information, state estimation becomes a critical component of output feedback MPC. In this study, Kalman filtering (Kailath, Sayed, & Hassibi, 2000) is employed to estimate the unmeasured states. By fusing the available sensor data with the system model, the Kalman filter generates an optimal estimate of the true state, enabling the MPC scheme to make informed and precise control decisions in the presence of limited and noisy information. Our designed and implemented filter uses the LPV model and therefore is parameter-dependent.

The Kalman filter operates in two main steps. In the prediction step, the system's dynamics model is used to predict the next state based on the previous state estimate and the control input, taking into account the system's inherent uncertainty as

$$\hat{x}^-_{k+1} = A(\mathrm{p}_k)\hat{x}_k + B(\mathrm{p}_k)u_k, \tag{4a}$$

$$\mathrm{P}^-_{k+1} = A(\mathrm{p}_k)\mathrm{P}_k A(\mathrm{p}_k)^T + \mathcal{Q}, \tag{4b}$$

where $\hat{x}^-_{k+1}$ is a priori state estimate at time instant $k+1$, and $\mathrm{P}^-_{k+1} = \mathbb{E}\left[(x_{k+1}-\hat{x}^-_{k+1})(x_{k+1}-\hat{x}^-_{k+1})^T | y_{0:k}\right]$ and $\mathrm{P}_k = \mathbb{E}\left[(x_k-\hat{x}_k)(x_k-\hat{x}_k)^T | y_{0:k}\right]$ denote the covariance of the estimation errors.

In the correction or update step, the filter incorporates the actual sensor measurements, along with their associated noise characteristics, to correct the predicted state estimate as

$$\hat{x}_{k+1} = \hat{x}^-_{k+1} + \mathrm{L}_{k+1}(y_{k+1} - C(\mathrm{p}_k)\hat{x}^-_{k+1}), \tag{5a}$$

$$\mathrm{P}_{k+1} = (I - \mathrm{L}_{k+1}C(\mathrm{p}_k))\mathrm{P}^-_{k+1}, \tag{5b}$$

$$\mathrm{L}_{k+1} = \mathrm{P}^-_{k+1}C(\mathrm{p}_k)^T\left(C(\mathrm{p}_k)\mathrm{P}^-_{k+1}C(\mathrm{p}_k)^T + \mathcal{R}\right)^{-1}. \tag{5c}$$

The Kalman filter dynamically adjusts the gain, denoted by L_{k+1} (which is parameter-varying), based on their respective uncertainties, optimizing the accuracy of the final state estimate.

The initial points of these iterative formulas are determined based on the known statistics of the initial condition in (2). If the modeling uncertainty exhibits a slow-varying nature, its temporal evolution can be disregarded, allowing the Kalman update to improve the estimation at each sampling time by incorporating the available measurements.

The accuracy of state estimation significantly impacts control performance, making it a critical aspect of the control system's effectiveness. Various methodologies have been explored to mitigate the impact of estimation errors, as evidenced by studies such as Subramanian, Lucia, and Engell (2014, 2018). Our future research will delve deeper into investigating the implications of estimation errors on control performance, aiming to develop strategies that enhance robustness and improve overall system performance.

4. Scenario-based model predictive control

In this section, a scenario-based model predictive control strategy is devised for the combustion control of RCCI engines. By leveraging the dynamic model, MPC predicts and optimizes future system behavior in real-time, while taking constraints into account to achieve optimal control performance within a finite time horizon. However, traditional MPC may encounter challenges when dealing with uncertainties and disturbances. To overcome this limitation, ScMPC expands the capabilities of MPC by explicitly incorporating uncertainty characteristics. This is achieved by considering a set of representative scenarios that encompass the most likely candidates for the system behavior. By evaluating control decisions against these scenarios, ScMPC offers a safer and more robust control strategy.

4.1. Optimal control problem formulation

The finite-horizon output feedback MPC problem under uncertainty is formulated as

$$\min_{\Theta} J(x,\vartheta,k) := \mathbb{E}\left\{\sum_{i=0}^{N_p-1} \ell_s(x_{i|k},\vartheta_{i|k}) + \ell_t(x_{N_p|k})\right\}, \tag{6a}$$

$$\text{s.t. } x_{i+1|k} = A(\mathrm{p}_k)x_{i|k} + B(\mathrm{p}_k)\vartheta_{i|k} + \tilde{w}_{i|k}, \tag{6b}$$

$$y_{i|k} = C(\mathrm{p}_k)x_{i|k} + v_{i|k}, \tag{6c}$$

$$x_{0|k} = \hat{x}_k, \tag{6d}$$

$$x \in \mathcal{X},\ \vartheta \in \mathcal{U},\ y \in \mathcal{Y}, \tag{6e}$$

where the decision variables are defined by $\Theta := \{\vartheta_{0|k}(.), ..., \vartheta_{N_p-1|k}(.)\}$, comprising a sequence of control laws $\vartheta_{i|k}$ over the prediction horizon N_p. Moreover, $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, $\mathcal{U} \subseteq \mathbb{R}^{n_u}$, and $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$ are the constraint sets of the states, inputs and outputs, respectively. $\mathbb{E}$ represents the expected value over the random vector sequence $\mathrm{W} = \{\tilde{w}_{0|k}, \dots, \tilde{w}_{N_p-1|k}\}$. Furthermore, $\ell_s(.)$ and $\ell_t(.)$ denote the stage and terminal cost functions, respectively.

Directly solving the optimization problem (6) is challenging. The main difficulty arises from the fact that optimizing over the control policy Θ yields an infinite-dimensional problem. To overcome this challenge, a practical approach is to approximate the control policy with an affine estimated state feedback policy with a fixed control gain, such as $\vartheta_{i|k} = u_{i|k} + K(x - \bar{x}_{i|k})$, where $\bar{x}_{i|k}$ represents the nominal state, $u_{i|k}$ is an auxiliary input, and K is a constant control gain. Now, the sequence $U = \{u_{0|k}, \dots, u_{N_p-1|k}\}$ is optimized using the nominal dynamic model in the optimization process.

Furthermore, representing uncertainty as a continuous random variable provides a more accurate model. Nevertheless, in practical scenarios, dealing with continuous probability distributions can be computationally demanding. To address this issue, a solution is to discretize the uncertainty, resulting in n_s discrete realizations of uncertainties at each time instant (de la Penad, Bemporad, & Alamo, 2005). Subsequently, a scenario tree is constructed, as shown in Fig. 1, to represent the evolution of uncertainty over the prediction horizon. Each path from the root node to a leaf node in the scenario tree corresponds to a specific scenario. However, the exponential growth in the number of scenarios due to branching at each time step leads to a substantial increase in the complexity of the optimization problem. A practical solution is to limit the branching by defining a *robust horizon* N_r, beyond which the number of uncertain scenarios are treated as fixed parameters (Lucia, Finkler, & Engell, 2013). Then, the problem will be limited to $n_s = n_d^{N_r}$ scenarios. This strategy helps in reducing the computational burden and facilitating the optimization process in the ScMPC framework.

The rigorous enforcement of hard constraints can often give rise to feasibility issues. The task of securing a feasible solution that adheres to all hard constraints for each scenario becomes increasingly demanding, particularly with the escalation in the number of scenarios or the extension of the robust and prediction horizon. This complexity arises from the expansion of uncertain scenarios over the prediction horizon, which in turn causes the invariant set to contract. Such a contraction can persist to the extent that it becomes impossible to identify a feasible solution. Additionally, even in cases where infeasibility does not arise, strictly adhering to hard constraints often leads to conservative control actions, which can constrain the system performance. Soft constraints, on the other hand, provide a more flexible and practical approach by allowing controlled violation of constraints. This flexibility enhances the robustness and feasibility of the control strategy, as it enables the MPC algorithm to generate feasible control solutions (Kerrigan & Maciejowski, 2000; Zhang & Morari, 1994).

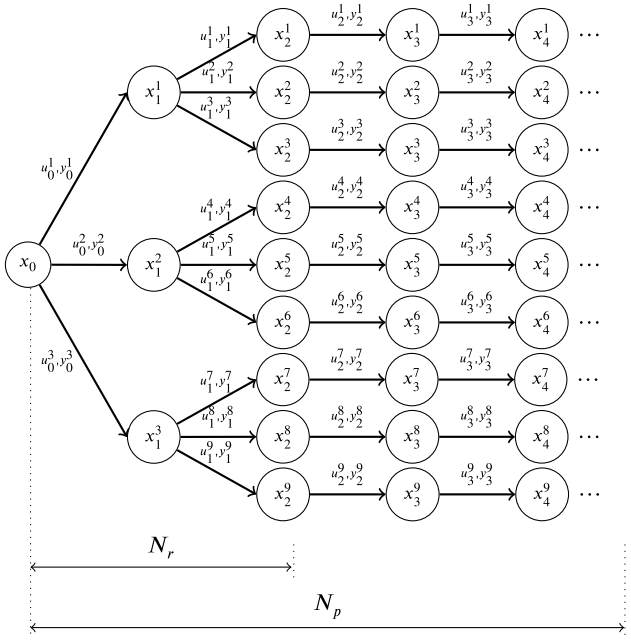


Fig. 1. The illustrative example shows a scenario tree with $n_d = 3$ discrete realizations of the uncertainty and a robust horizon of $N_r = 2$.

To formulate the tracking problem in the ScMPC framework and incorporate an integral action for precise performance, we augment the state-space model as

$$\mathrm{x}_{k+1} = \tilde{A}_k \mathrm{x}_k + \tilde{B}_k \Delta u_k + \Omega_k, \tag{7a}$$

$$y_k = \tilde{C}_k \mathrm{x}_k + \upsilon_k, \tag{7b}$$

$$\tilde{A}_k = \begin{bmatrix} A(\mathrm{p}_k) & B(\mathrm{p}_k) \\ 0 & I \end{bmatrix}, \quad \tilde{B}_k = \begin{bmatrix} B(\mathrm{p}_k) \\ I \end{bmatrix},$$

$$\tilde{C}_k = \begin{bmatrix} C(\mathrm{p}_k) & 0 \end{bmatrix}, \quad \Omega_k = \begin{bmatrix} \tilde{\omega}_k^T & 0 \end{bmatrix}^T,$$

where the augmented state vector x_k is $\mathrm{x}_k = [x_k^T \;\; u_{k-1}^T]^T$. Additionally, Δu_k represents the change in the control input from the previous time step, defined as $\Delta u_k = u_k - u_{k-1}$. Therefore, the output feedback scenario-based optimal tracking control problem for an uncertain system, incorporating penalties for constraint violations, ϵ, is formulated as

$$\min_{y,u,\epsilon} \sum_{j=1}^{n_s} \pi^j \Bigg[\sum_{i=0}^{N_p-1} \|y_{i|k}^j - r_k\|_Q^2 + \|\Delta u_{i|k}^j\|_R^2 + \|\epsilon_{i|k}^j\|_S^2 + \|y_{N_p|k}^j - r_k\|_P^2 + \|\epsilon_{N_p|k}^j\|_S^2 \Bigg], \tag{8a}$$

$$\text{s.t.} \quad \mathrm{x}_{i+1|k}^j = \tilde{A}_k \mathrm{x}_{i|k}^j + \tilde{B}_k \Delta u_{i|k}^j + \Omega_{i|k}^j, \tag{8b}$$

$$y_{i|k}^j = \tilde{C}_k \mathrm{x}_{i|k}^j + \upsilon_{i|k}^j, \tag{8c}$$

$$\mathrm{x}_{0|k} = \begin{bmatrix} \hat{x}_k^T & u_{k-1} \end{bmatrix}^T, \tag{8d}$$

$$G_{\mathrm{x}} \, \mathrm{x}_{i|k}^j \le g_{\mathrm{x}}, \tag{8e}$$

$$G_u \, \Delta u_{i|k}^j \le g_u, \tag{8f}$$

$$G_y \, y_{i|k}^j \le g_y + \epsilon_{i|k}^j, \tag{8g}$$

$$\Delta u_{i|k}^j = \Delta u_{i|k}^l \text{ if } \mathrm{x}_{i|k}^{p_j} = \mathrm{x}_{i|k}^{p_l}, \tag{8h}$$

where the superscript $j \in \{1, \dots, n_s\}$ designates the scenario number, and the subscript $i|k$, $i \in \{0, \dots, N_p - 1\}$, denotes the i-step-ahead prediction of variables initiated at time instant k. Polytopic constraints on augmented states, inputs variation, and outputs are represented by $G_{\mathrm{x}} \in \mathbb{R}^{(m_{\mathrm{x}}+m_u)\times(n_{\mathrm{x}}+n_u)}, g_{\mathrm{x}} \in \mathbb{R}^{(m_{\mathrm{x}}+m_u)}$, $G_u \in \mathbb{R}^{m_u \times n_u}, g_u \in \mathbb{R}^{m_u}$, and $G_y \in \mathbb{R}^{m_y \times n_y}, g_y \in \mathbb{R}^{m_y}$. Additionally, $Q \succeq 0$, $R \succ 0$, $P \succ 0$, and $S \succ 0$ denote weighting matrices, while π^j signifies the probability of the jth scenario. The reference trajectory to be tracked is denoted by r_k.

The *non-anticipativity constraint*, (8h), enforces the condition that all decisions branching from the same parent node, $\mathrm{x}_{i|k}^{p_{(.)}}$, in the scenario tree must be the same. This reflects the fact that the control inputs cannot anticipate or depend on the future realization of uncertainty. By imposing non-anticipativity, the MPC algorithm ensures that control decisions are made based on the available information at each time instant without any knowledge of future uncertainties (Goodwin, Østergaard, Quevedo, & Feuer, 2009).

The solution to the optimization problem (8) is denoted by $U_k^*(\hat{x}_k)$ and the control law is defined as the first element of the row corresponding to the highest probability, i.e.,

$$u_k = u_{0|k}^{j^*}. \tag{9}$$

4.2. Learning-based scenario generation

The ScMPC relies on a limited number of scenarios representing different realizations of uncertainties and disturbances. These scenarios should be carefully selected to capture the range of possible variations in the system's behavior due to the propagation of uncertainties within the prediction horizon. Unlike related works in the literature that primarily focus on input- and state-dependent uncertainty quantification, this work proposes to learn input- and output-dependent uncertainties, which offers a more practical approach to the problem at hand and facilitates output feedback control design. To achieve this, the scenarios are generated using a BNN model to represent the uncertainties associated with the LPV model identification, as presented in Bao, Velni, and Shahbakhti (2020). At each time instant, the BNN model evaluates a stochastic distribution of the uncertainty based on the control input, scheduling variables and plant outputs, and then three scenarios are defined that provide a multifaceted view of the system's potential behaviors, enabling the ScMPC controller to consider and adapt to different outcomes.

In the BNN framework, the parameters of the neural network are treated as random variables and assigned prior distributions as

$$p(\mathrm{w}_j) = \rho_{\mathrm{m},j} \mathcal{N}(\mathrm{w}_j | 0, \sigma_{j,1}^2) + (1 - \rho_{\mathrm{m},j}) \mathcal{N}(\mathrm{w}_j | 0, \sigma_{j,2}^2). \tag{10}$$

The tuning parameter, $\rho_{\mathrm{m},j}$, serves as the controlling factor for the prior density of the parameters in the jth layer of the neural network, denoted as w_j. It influences the shape and characteristics of the prior distribution, allowing for flexible modeling of uncertainty in the network's parameters.

During the training process, the network undergoes learning to approximate the posterior distribution of its parameters, given the available dataset $\mathcal{T}$, by leveraging variational inference (VI) techniques with

$$\mathcal{T} = \left\{ \mathrm{u}^{(i)} = (y^{(i)}, u^{(i)}, \mathrm{p}^{(i)}), \mathfrak{y}^{(i)} = \tilde{w}^{(i)} \right\}_{i=1}^{n_t}, \tag{11}$$

where $\mathrm{u}^{(i)}$ and $\mathfrak{y}^{(i)}$ are the ith input and output of the network, and n_t is the number of training data points. VI approximates complex probability distributions by selecting a member from a family of densities that closely approximates the target distribution, minimizing the Kullback–Leibler (KL) divergence between the true posterior distribution and the approximation (Blei, Kucukelbir, & McAuliffe, 2017). To approximate the posterior $p(\mathrm{w}_j|\mathcal{T})$, VI solves

$$\min_{\theta_j} \mathrm{KL}\Big(q(\mathrm{w}_j; \theta_j) \parallel p(\mathrm{w}_j|\mathcal{T}) \Big) \tag{12a}$$

$$\Leftrightarrow \quad \min_{\theta_j} \mathrm{KL}\Big(q(\mathrm{w}_j; \theta_j) \parallel p(\mathrm{w}_j) \Big) - \mathbb{E}_{q(\mathrm{w}_j;\theta_j)}[\log p(\mathrm{w}_j|\mathcal{T})]$$

$$\Leftrightarrow \quad \min_{\theta_j} \Big(\mathbb{E}_{q(\mathrm{w}_j;\theta_j)}[\log q(\mathrm{w}_j; \theta_j)] - \mathbb{E}_{q(\mathrm{w}_j;\theta_j)}[\log p(\mathrm{w}_j)]$$

$$- \mathbb{E}_{q(\mathrm{w}_j;\theta_j)}[\log p(\mathcal{T}|\mathrm{w}_j)] \Big), \tag{12b}$$

where $q(\mathrm{w}_j;\theta_j)$ represents a family of densities with parameters θ_j. The evidence lower bound (ELBO) function, (12b), is solved by Monte Carlo (MC) methods and backpropagation (Blei et al., 2017), which parameterizes $q(\mathrm{w}_j;\theta_j)$ as $\mathrm{w}_j = \mu_j + \sigma_j \bigodot \varepsilon_j$, where $\bigodot$ denotes the element-wise multiplication, $\varepsilon_j \sim \mathcal{N}(0, I)$, and so $\theta_j = (\mu_j, \sigma_j)$. Then, the BNN is trained by solving the following problem over the dataset $\mathcal{T}$

$$\min_{\theta} \frac{1}{n_b} \sum_{i=1}^{n_b} [\log q(\mathrm{w}^{(i)};\theta) - \log p(\mathrm{w}^{(i)}) - \log p(\mathcal{T}|\mathrm{w}^{(i)})], \tag{13}$$

where $\mathrm{w}^{(i)}$ is the ith sample generated by MC for approximating the ELBO, and n_b is the MC sample size determined such that (13) converges to a local optimum. To ensure that the estimated mismatch is reliable and maintains the system inside its operating region, the following assumption is made:

Assumption 1. For a confidence level $\delta \in (0, 1]$, there exists a scaling factor α such that with a probability greater than $1 - \delta$,

$$\forall k \in \mathbb{N},\ |\tilde{w}_k^{\iota} - \hat{\mu}_k^{\iota}| \le \alpha_{\iota} \hat{\sigma}_k^{\iota} < |\mathcal{W}_{\iota}|, \iota \in \{1, \dots, n_x\} \tag{14}$$

given $(y_k, u_k) \in \mathcal{Y} \times \mathcal{U}$, where $\hat{\mu}_k^{\iota}$ and $\hat{\sigma}_k^{\iota}$ denote the estimated mean and standard deviation of the ιth entry of $\tilde{w}_k$, respectively, using the learned BNN model with MC methods, and $|\mathcal{W}_{\iota}|$ denotes the maximum valid value of $\tilde{w}_k^{\iota}$.

By adhering to Assumption 1, we guarantee that the trained BNN model achieves sufficient accuracy, ensuring that the predicted mismatch values lie within the credible intervals outlined by our probabilistic model. In cases where Assumption 1 is not satisfied by the trained model, enhancing the model should involve modifying the architecture and optimization techniques, or enriching the training dataset until the prescribed criteria are fulfilled.

At each time instant k, a total of n_{MC} samples are drawn from normal distributions, which are then used to calculate the weights $\mathrm{w}^{(i)}$ by applying the reparameterization trick to each individual sample. As the ScMPC works with a finite number of scenarios, instead, similar to Bao et al. (2023), we calculate

$$\hat{\mu}_k = \frac{1}{n_{\mathrm{MC}}} \sum_{i=1}^{n_{\mathrm{MC}}} \tilde{w}^{(i)}, \tag{15}$$

$$\hat{\sigma}_k = \sqrt{\frac{1}{n_{\mathrm{MC}}} \sum_{i=1}^{n_{\mathrm{MC}}} (\tilde{w}^{(i)} - \hat{\mu}_k)^{\top} (\tilde{w}^{(i)} - \hat{\mu}_k)}\ , \tag{16}$$

and use $\hat{\mu}_k,\ \hat{\mu}_k \pm \alpha^j \hat{\sigma}_k,\ j = 1, \dots, \frac{n_d - 1}{2}$ as the scenarios at each node of a stage in the scenario tree. Here, α^j's are the tuning multipliers for n_d discrete uncertainty realizations.

4.3. Probability of the scenarios

After generating the scenario tree, the probability of each scenario is calculated using the moment matching method. Specifically, the first four central moments, including mean, variance, skewness and kurtosis, are matched. This approach ensures that the generated probabilities accurately reflect the underlying distribution and preserve key statistical characteristics. To achieve this, the following optimization problem is solved:

$$\begin{aligned}
\min_{\mathbf{\Pi}} \ & \sum_{i}^{n_w} \Big(c_i^1 (M_i^- + M_i^+) + c_i^3 (\Psi_i^- + \Psi_i^+) \\
& + c_i^4 (\Gamma_i^- + \Gamma_i^+) + \sum_{i,j=1}^{n_w} c_{i,j}^2 (\Sigma_{i,j}^- + \Sigma_{i,j}^+) \Big), \\
\text{s.t. } & \mathbf{X}\,\mathbf{\Pi} + M_i^- + M_i^+ = M, \\
& \sum_{i=1}^{n_s} \left(\mathbf{X}^i - \mathbf{X}\,\mathbf{\Pi}\right)^2 \pi^i + \Sigma^- - \Sigma^+ = \Sigma, \\
& \sum_{i=1}^{n_s} \left(\mathbf{X}^i - \mathbf{X}\,\mathbf{\Pi}\right)^3 \pi^i + \Psi^- - \Psi^+ = \Psi, \\
& \sum_{s=1}^{n_s} \left(\mathbf{X}^i - \mathbf{X}\,\mathbf{\Pi}\right)^4 \pi^i + \Gamma^- - \Gamma^+ = \Gamma, \\
& \sum_{i=1}^{n_s} \pi^i = 1,\ \pi^i \ge 0,\ i = 1, \dots, n_s, \\
& M_i^-, M_i^+, \Psi_i^-, \Psi_i^+, \Gamma_i^-, \Gamma_i^+ \ge 0,\ i = 1, \dots, n_w, \\
& \Sigma_{i,j}^-, \Sigma_{i,j}^+ \ge 0,\ i, j = 1, \dots, n_w.
\end{aligned} \tag{17}$$

Here, the notation $\left(\mathbf{X}^i - \mathbf{X}\,\mathbf{\Pi}\right)^n$ represents the nth central moment, while M, Σ, Ψ, and Γ denote the estimated first four central moments obtained from the samples. The superscripts + and − indicate the positive and negative parts of the associated variables. The objective function incorporates the weighting coefficients c_i^1, c_{ij}^2, c_i^3, and c_i^4. Moreover, $\mathbf{\Pi} = (\pi^1, \dots, \pi^{n_s})^T$ represents the vector of probabilities for each scenario, $\pi^i, i = 1, \dots, n_s$. The matrix $\mathbf{X} = (\mathbf{X}^1, \dots, \mathbf{X}^{n_s}) \in \mathbb{R}^{n_w \times n_s}$ consists of the uncertainty realizations, with $\mathbf{X}^i = (X_1^i, \dots, X_{n_w}^i)$ representing the realization of the uncertainty for the ith scenario. Here, n_w denotes the dimension of each realization and for this problem, it is equal to the dimension of state vector, n_x. Finally, the number of scenarios, n_s, is determined such that the cost value of the optimization problem remains within an acceptable range. Next, using the generated adaptive scenario tree described in Section 4.2 and the scenario probability calculated from the optimization problem (17), at each time instant, the optimal control problem (8) can be solved and the control law (9) can be obtained.

5. Results and discussion

The architecture of the proposed combustion control scheme for RCCI engines is shown in Fig. 2. To investigate the effectiveness of the proposed scenario-based output feedback MPC for RCCI engines, extensive simulations were conducted using a high-fidelity experimentally validated engine model (Raut, Irdmousa and Shahbakhti, 2018) which captures the intricate dynamics of RCCI combustion and fuel injection, enabling a realistic representation of the engine behavior under different operating conditions. This model was built to represent the dynamics of a modified GM engine, a 2.0 L, 4-cylinder Ecotec turbocharged gasoline direct injection engine, enhanced to facilitate dual fuel operation in RCCI mode. The engine uses n-heptane as the high reactivity fuel and iso-octane as the low reactivity fuel. The direct fuel injection pressure is regulated at 100 bar, whereas the port fuel injection operates at a fuel pressure of 3 bar (Raut, Irdmousa et al., 2018).

The initial phase of the study involved learning a five-state LPV representation of the RCCI engine using the least-squares support vector machines (LS-SVM) (Rizvi, Velni, Tóth, & Meskin, 2015; Sitaraman et al., 2022). In order to generate data for system identification, open-loop simulations were run on the engine model in which the variables of interest, i.e., (3a), were manipulated as shown in Fig. 3(a), and the resulting states of the system were recorded as shown in Fig. 3(c). The scheduling variables, which are functions of inputs, follow the trend shown in Fig. 3(b). Throughout the simulations, the engine speed was consistently maintained at 1000 rpm, and the intake manifold temperature and pressure were $T_m = 333.15$ K and $P_m = 96.5$ kPa, respectively. This data were then fed into the SVM-based model learning algorithm to generate the LPV state-space matrices (Sitaraman et al., 2022).

By comparing the states obtained from the high-fidelity model and the learned LPV model, the mismatch is determined, as shown in Fig. 4. The training dataset, represented as (11), was created using these signals, consisting of 927 samples. Subsequently, it was randomly partitioned into training and testing datasets, with a split ratio of 67%

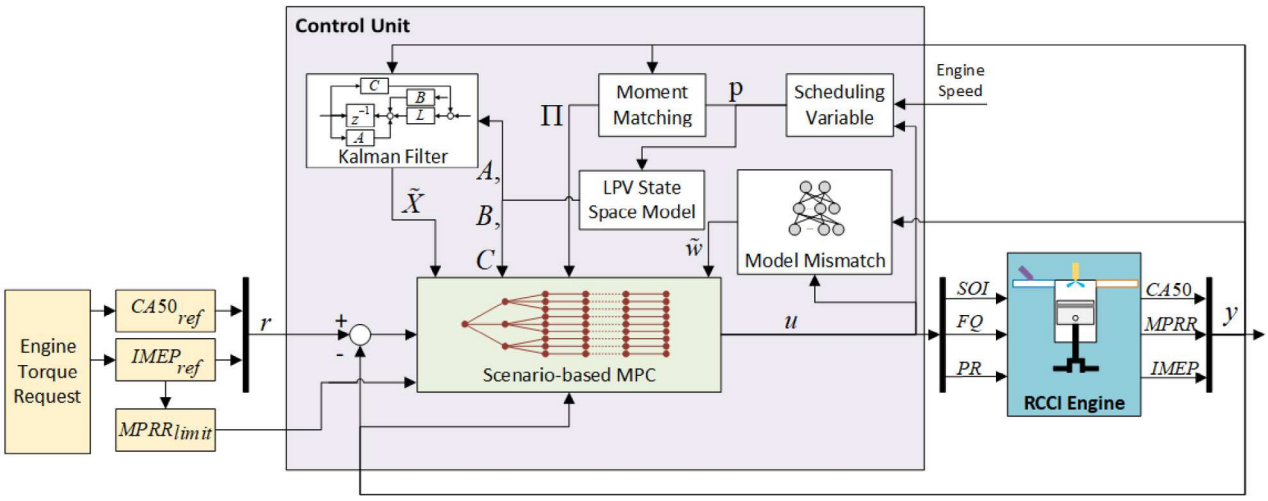


Fig. 2. Block diagram illustrating the proposed combustion controller. At each time step, the scenario-tree is constructed utilizing estimated states, LPV matrices, discrete scenarios derived from the plant-model mismatch, and their associated probabilities.

Table 1
Input and output constraints of the RCCI engine.

Variable	Unit	Min. limit	Max. limit
SOI	CAD bTDC	32	45
FQ	mg/cycle	18	27
PR	%	0	40
$CA50$	CAD aTDC	−10	30
$IMEP$	kPa	500	1000
$MPRR$	bar/CAD	0	6

for training and 33% for testing. The BNN model was constructed with a three-layer fully connected neural network, which is subsequently connected to a DenseVariational layer featuring linear activation. The neural network employs Exponential Linear Unit (ELU) activation functions. Notably, each of the hidden layers in the neural network consists of 16 units. The hyperparameters of the prior (10) are set as $\rho = 0.5$, $\sigma_1 = 1$, and $\sigma_2 = 0.1$. The BNN was trained using the Adam optimizer implemented in Keras with the learning rate set to 10^{-5} and decay to 10^{-6}. All other parameters of Adam were kept at their default values. The model was trained for a total of 5000 epochs with a batch size of 8. The results show that 99.03% of training samples and 95.75% of testing samples fall within the confidence interval defined as $[\hat{\mu}-2\hat{\sigma}, \hat{\mu}+2\hat{\sigma}]$. The validation results for the BNN model on the testing data are presented in Fig. 5, which confirms the high accuracy of the model.

For online scenario generation within the feedback control loop, n_{MC} samples are drawn at each engine cycle to estimate the mean, $\hat{\mu}_k$, and standard deviation, $\hat{\sigma}_k$, of the mismatch. The parameter n_{MC} is determined as the minimum number of sampled models required to satisfy a given confidence interval based on the design requirements. We initially made an educated guess for n_{MC} and fine-tuned the parameter through iterative adjustments until we converged on $n_{\text{MC}} = 50$ (Bao et al., 2023). Subsequently, the probabilities of the scenarios, illustrated in Fig. 6, are computed using the moment matching method, achieved by solving (17).

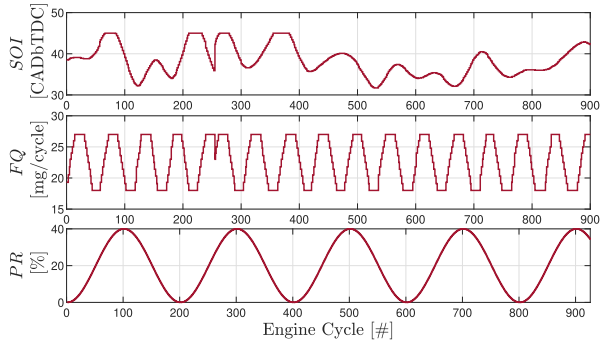
The primary control objective is to ensure that both $CA50$ and $IMEP$ closely follow the desired trajectories, determined from the engine speed and torque request, while keeping the engine's inputs and outputs within the specified constraint range as presented in Table 1. The prediction horizon is set to 20 engine cycles, while the robust horizon is chosen to be 2 engine cycles. For a more realistic scenario, the output measurements are assumed to be corrupted with noise derived from actual experimental data, with the covariance matrix $\mathcal{R} = \text{diag}([1, 0.7, 28])$.

The tracking capability of the proposed controller is illustrated in Fig. 7 and compared against both the baseline MPC (Sitaraman et al., 2022), as well as the worst-case scenario MPC. When focusing on $CA50$ tracking, it is evident that the proposed ScMPC swiftly follows the reference with remarkable precision, displaying a minimal root-mean-square error (RMSE) of only 1.61 aTDC. In stark contrast, the baseline MPC exhibits a delay of approximately eight engine cycles after cycle #80, resulting in an error of 3.79 aTDC. Furthermore, it demonstrates reduced fluctuations compared to the worst-case scenario, yielding a tracking error of just 2.32 aTDC. When it comes to $IMEP$ tracking, once again, the ScMPC outperforms other control schemes by showcasing the smallest RMSE of 29.88 kPa. Regarding $MPRR$, the ScMPC exhibits only a slight deviation of 0.27 bar/CAD, whereas the baseline MPC and the worst-case scenario show higher deviations of 1.05 bar/CAD and 1.17 bar/CAD, respectively.

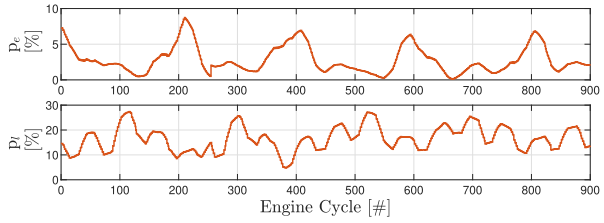
The control inputs of three control schemes are compared in Fig. 8. As expected, all control inputs consistently operate within the prescribed hard constraints, underscoring the effectiveness of the controllers in maintaining system stability and adhering to safety limits. During the simulations, it becomes evident that the total amount of fuel injected by the three controllers remains nearly identical, with values of approximately 0.51 g for both the baseline and scenario-based MPC and slightly higher at 0.53 g for the worst-case scenario. Notably, there is a discernible observation that as the combustion phasing is retarded between cycles #60 and #80, the ScMPC injects less low reactivity fuel to achieve control objectives.

The scenario-based optimization problem under consideration presents a remarkable challenge, primarily stemming from the tight output constraints and the proximity of the IMEP reference to those boundaries. As previously highlighted, the primary objective of the ScMPC is to enhance the system's robustness against modeling uncertainty. As the scenario tree is constructed over the prediction horizon, the propagation of uncertainty inherent in the data-driven model becomes evident. This uncertainty results in deviations from the nominal model, leading to considerable divergence of the furthest leaf nodes, representing extreme scenarios. The extent of this divergence is observed from Fig. 9, highlighting the potential for scenarios to deviate from the nominal model and violate constraints. While not necessarily reflective of real-world operating conditions, these scenarios serve as vital inputs for the ScMPC controller, making it robust against modeling uncertainty.

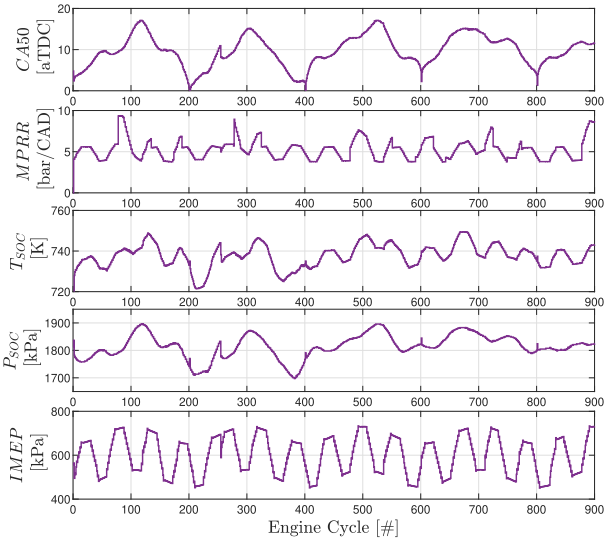
When applying hard constraints on output variables, the MPC optimization problem faces a delicate trade-off between tracking performance and constraint adherence, provided that the optimization



(a) Inputs to the engine.



(b) Scheduling variables of the engine.



(c) States of the engine.

Fig. 3. Model identification inputs, scheduling variables and states at engine speed $N = 1000$ rpm, $T_m = 333.15$ K and $P_m = 96.5$ kPa.

problem is feasible. This compromise leads to a degradation in tracking performance. However, soft constraints can be employed as a valuable solution, allowing outer branches and less probable scenarios to temporarily breach the hard constraints, while still keeping these deviations minimized. This strategic approach ensures robust tracking performance even in the presence of uncertainties, while maintaining feasibility in solving the optimization problem.

Combustion stability can be assessed using the coefficient of variation of IMEP defined as

$$COV_{IMEP} = \frac{\sigma_{IMEP}}{\mu_{IMEP}} \tag{18}$$

where μ_{IMEP} and σ_{IMEP} denote the mean and standard deviation, respectively. Operational zones with COV_{IMEP} values below 5% are classified as stable operating regions which not only ensures low emissions, but also reduces engine speed fluctuations and favorable NVH characteristics (Raut, Bidarvatan et al., 2018). In our study, this index is

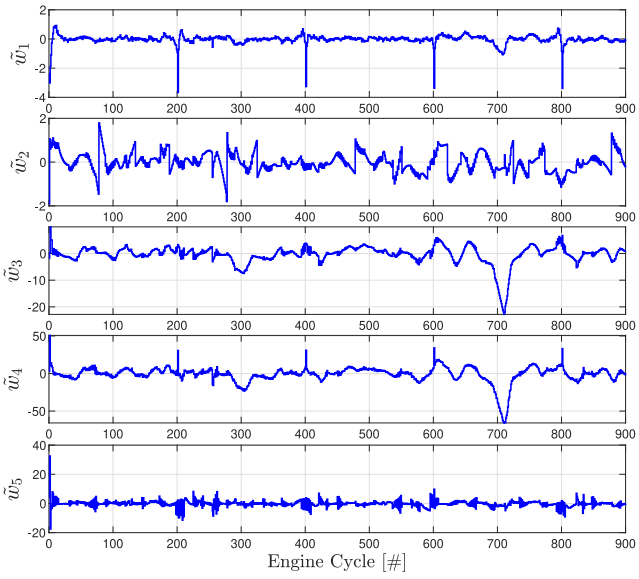


Fig. 4. RCCI engine model mismatch.

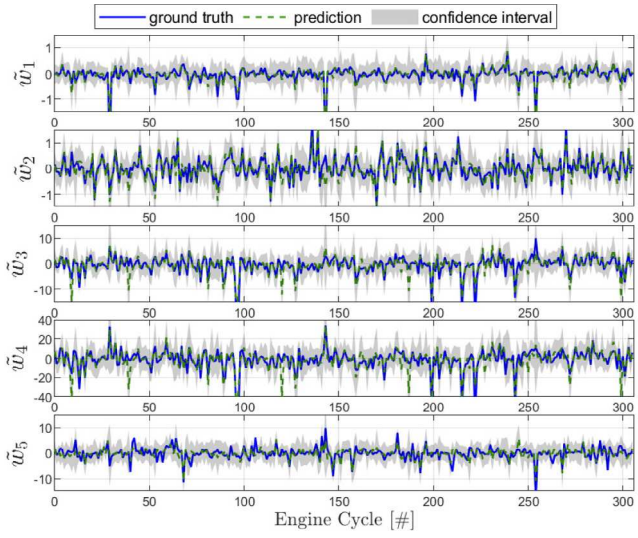


Fig. 5. Comparison between the testing data and predicted mean value of mismatch (using BNN) in the confidence interval.

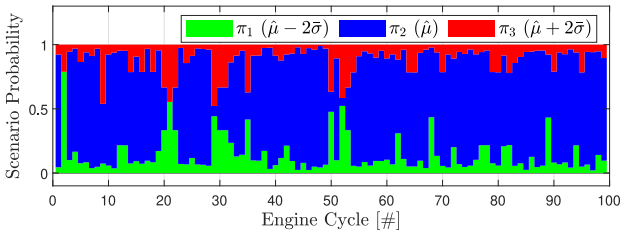


Fig. 6. Probability of the scenarios.

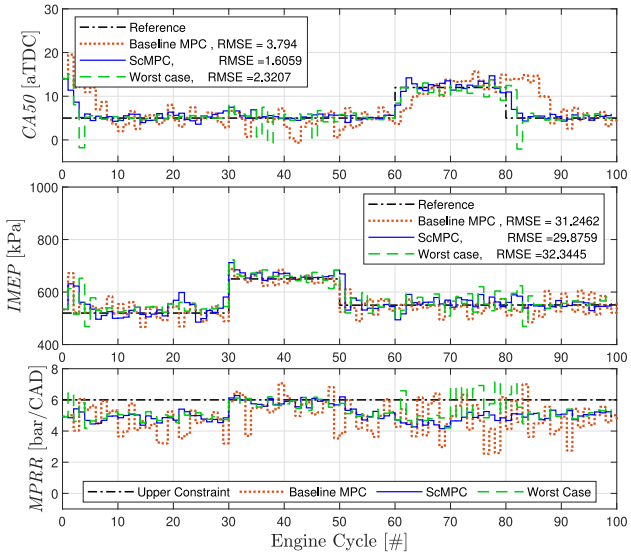


Fig. 7. Tracking performance of the proposed ScMPC controller compared with baseline MPC controller (Sitaraman et al., 2022) and the worst case scenario.

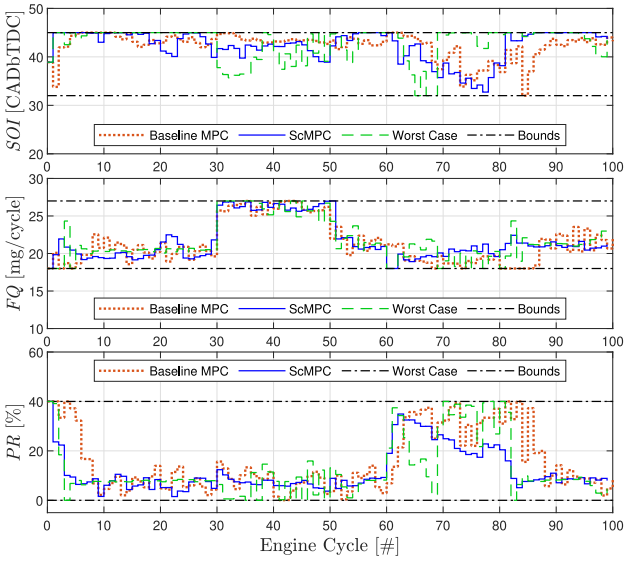


Fig. 8. Control inputs of the three designed controllers bounded by constraints.

calculated over five engine cycles and presented in Fig. 10. As observed, the controller successfully maintains the combustion stability.

6. Conclusion

This paper presents a new approach to output feedback scenario-based model predictive control design for high performance combustion control of RCCI engines. The proposed controller demonstrates robust performance by accounting for measurement noise and modeling uncertainties, effectively handling real-world complexities. For dynamic modeling, LS-SVM is employed to map the complex dynamics of the engine combustion process into an LPV description. This learning-based approach proves effective in accurately representing the engine's intricate behavior. To address stochastic modeling uncertainty, Bayesian Neural Networks (BNN) is utilized, enabling the quantification of uncertainties based on input and output data. This BNN model is then used for online scenario generation in the feedback control loop.

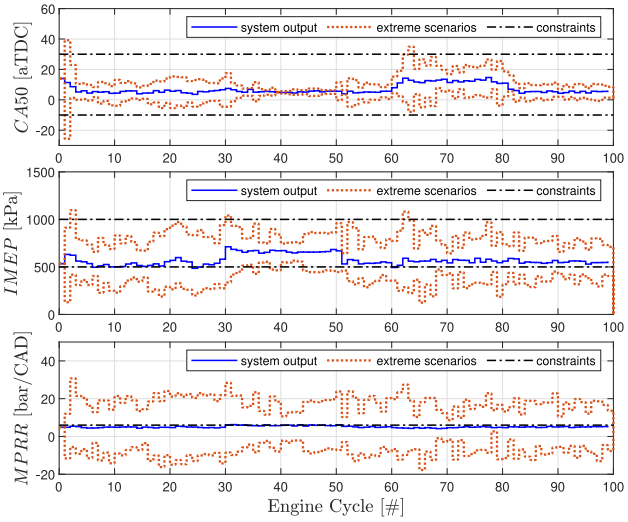


Fig. 9. Extreme scenarios in contrast to constraints.

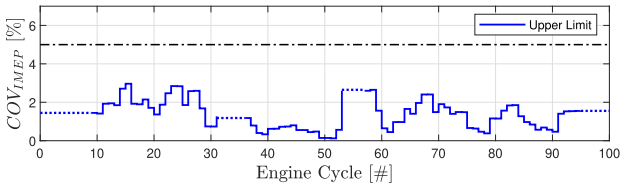


Fig. 10. Coefficient of variation of $IMEP$ compared to combustion stability limit.

The moment-matching technique is subsequently employed to calculate probabilities for the generated scenarios, empowering the controller to make realistic decisions. To ensure the feasibility of the underlying predictive optimization problem, soft constraints are applied to the output constraints, providing more flexibility in the control process. Furthermore, as the engine states are not directly measurable, a parameter-dependent Kalman filter is used for state estimation, enabling the controller to work with estimated state information. Simulation results validate the efficacy of the proposed controller in handling challenges posed by noise and uncertainties while effectively controlling constraint violations on outputs. The ScMPC strategy emerged as a promising solution for real-time control of combustion in RCCI engines, offering improved stability and performance compared to conventional control methods. Our future work will focus on considering the state estimation error in the output feedback MPC-LPV design.

CRediT authorship contribution statement

Pegah GhafGhanbari: Data curation, Formal analysis, Writing – original draft. **Yajie Bao:** Data curation, Writing – review & editing. **Javad Mohammadpour Velni:** Funding acquisition, Supervision, Writing – review & editing.

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.

References

Agarwal, A. K., Singh, A. P., & Maurya, R. K. (2017). Evolution, challenges and path forward for low temperature combustion engines. *Progress in Energy and Combustion Science, 61*, 1–56.

Bao, Y., Chan, K. J., Mesbah, A., & Velni, J. M. (2023). Learning-based adaptive-scenario-tree model predictive control with improved probabilistic safety using robust Bayesian neural networks. *International Journal of Robust and Nonlinear Control*, *33*(5), 3312–3333.

Bao, Y., Velni, J. M., & Shahbakhti, M. (2020). Epistemic uncertainty quantification in state-space LPV model identification using Bayesian neural networks. *IEEE Control Systems Letters*, *5*(2), 719–724.

Basina, L. A., Irdmousa, B. K., Velni, J. M., Borhan, H., Naber, J. D., & Shahbakhti, M. (2020). Data-driven modeling and predictive control of maximum pressure rise rate in RCCI engines. In *2020 IEEE conference on control technology and applications* (pp. 94–99). IEEE.

Batool, S., Naber, J. D., & Shahbakhti, M. (2021). Data-driven modeling and control of cyclic variability of an engine operating in low temperature combustion modes. *IFAC-PapersOnLine*, *54*(20), 834–839.

Bernardini, D., & Bemporad, A. (2009). Scenario-based model predictive control of stochastic constrained linear systems. In *Proceedings of the 48h IEEE conference on decision and control (CDC) held jointly with 2009 28th Chinese control conference* (pp. 6333–6338). IEEE.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, *112*(518), 859–877.

Bonzanini, A. D., Paulson, J. A., Makrygiorgos, G., & Mesbah, A. (2021). Fast approximate learning-based multistage nonlinear model predictive control using Gaussian processes and deep neural networks. *Computers & Chemical Engineering*, *145*, Article 107174.

Campo, P. J., & Morari, M. (1987). Robust model predictive control. In *1987 American control conference* (pp. 1021–1026). IEEE.

de la Penad, D. M., Bemporad, A., & Alamo, T. (2005). Stochastic programming applied to model predictive control. In *Proceedings of the 44th IEEE conference on decision and control* (pp. 1361–1366). IEEE.

Goodwin, G. C., Østergaard, J., Quevedo, D. E., & Feuer, A. (2009). A vector quantization approach to scenario generation for stochastic NMPC. In *Nonlinear model predictive control: towards new challenging applications* (pp. 235–248). Springer.

Irdmousa, B. K., Rizvi, S. Z., Velni, J. M., Nabert, J., & Shahbakhti, M. (2019). Data-driven modeling and predictive control of combustion phasing for RCCI engines. In *2019 American control conference* (pp. 1617–1622). IEEE.

Kailath, T., Sayed, A. H., & Hassibi, B. (2000). *Linear estimation*. Prentice Hall, BOOK.

Kerrigan, E. C., & Maciejowski, J. M. (2000). Soft constraints and exact penalty functions in model predictive control. In *Control 2000 conference, Cambridge* (pp. 2319–2327).

Krishnamoorthi, M., Malayalamurthi, R., He, Z., & Kandasamy, S. (2019). A review on low temperature combustion engines: Performance, combustion and emission characteristics. *Renewable and Sustainable Energy Reviews*, *116*, Article 109404.

Lu, X., Han, D., & Huang, Z. (2011). Fuel design and management for the control of advanced compression-ignition combustion modes. *Progress in Energy and Combustion Science*, *37*(6), 741–783.

Lucia, S., Finkler, T., & Engell, S. (2013). Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. *Journal of Process Control*, *23*(9), 1306–1319.

Mayne, D. Q., Seron, M. M., & Raković, S. (2005). Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, *41*(2), 219–224.

Mesbah, A. (2016). Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, *36*(6), 30–44.

Raut, A., Bidarvatan, M., Borhan, H., & Shahbakhti, M. (2018). Model predictive control of an RCCI engine. In *2018 annual American control conference* (pp. 1604–1609). IEEE.

Raut, A., Irdmousa, B. K., & Shahbakhti, M. (2018). Dynamic modeling and model predictive control of an RCCI engine. *Control Engineering Practice*, *81*, 129–144.

Rizvi, S. Z., Velni, J. M., Tóth, R., & Meskin, N. (2015). A kernel-based approach to MIMO LPV state-space identification and application to a nonlinear process system. *IFAC-PapersOnLine*, *48*(26), 85–90.

Singh, A. P., Kumar, V., & Agarwal, A. K. (2020). Evaluation of comparative engine combustion, performance and emission characteristics of low temperature combustion (PCCI and RCCI) modes. *Applied Energy*, *278*, Article 115644.

Sitaraman, R., Batool, S., Borhan, H., Velni, J. M., Naber, J. D., & Shahbakhti, M. (2022). Data-driven model learning and control of RCCI engines based on heat release rate. *IFAC-PapersOnLine*, *55*(37), 608–614.

Subramanian, S., Lucia, S., & Engell, S. (2014). Economic multi-stage output nonlinear model predictive control. In *2014 IEEE conference on control applications* (pp. 1837–1842). IEEE.

Subramanian, S., Lucia, S., & Engell, S. (2018). A synergistic approach to robust output feedback control: Tube-based multi-stage NMPC. 577–582. In *10th IFAC symposium on advanced control of chemical processes ADCHEM*.

Zhang, A., & Morari, M. (1994). Stability of model predictive control with soft constraints. *Vol. 2*, In *Proceedings of 1994 33rd IEEE conference on decision and control* (pp. 1018–1023). IEEE.