Control Engineering Practice 61 (2017) 206–216

Contents lists available at ScienceDirect

Control Engineering Practice

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

Optimal control for integrated emission management in diesel engines ☆

M.C.F. Donkers [a,*], J. Van Schijndel [b], W.P.M.H. Heemels [b], F.P.T. Willems [b,c]

[a] Department of Electrical Engineering, Eindhoven University of Technology, Netherlands
[b] Department of Mechanical Engineering, Eindhoven University of Technology, Netherlands
[c] TNO Automotive, Powertrains, Netherlands

ARTICLE INFO

Article history:
Received 31 May 2015
Received in revised form
30 November 2015
Accepted 6 March 2016
Available online 28 March 2016

Keywords:
Optimal control
Dynamic programming
Automotive control

ABSTRACT

Integrated Emission Management (IEM) is a supervisory control strategy that minimises operational costs (consisting of fuel and AdBlue) for diesel engines with an aftertreatment system, while satisfying emission constraints imposed by legislation. In most work on IEM, a suboptimal heuristic real-time implementable solution is used, which is based on Pontryagin's Minimum Principle (PMP). In this paper, we compute the optimal solution using both PMP and Dynamic Programming (DP). As the emission legislation imposes a terminal state constraint, standard DP algorithms are sensitive to numerical errors that appear close to the boundary of the feasible sets. Therefore, we propose two extensions to existing DP methods, which use an approximation of the forward reachable sets to reduce the grid size over time and an approximation of the backward reachable sets to avoid the aforementioned numerical errors. Using a simulation study of a cold-start World Harmonised Transient Cycle for a Euro-VI engine, we show that the novel extension to the DP algorithm yields the best approximation of the optimal cost, when compared to existing DP methods. Furthermore, we show that PMP yields almost the same results as DP, and that the real-time implementable solution only deviates approximately 0.08–0.16% from the optimal solution.

1. Introduction

Modern emission legislation forces tailpipe emissions of nitrogen oxides (NO_x) and particulate matter (PM) of heavy-duty trucks towards near-zero impact levels. Moreover, the current challenge set by the European Commission is to achieve a 20% fuel consumption reduction in heavy-duty trucks by 2020. To meet these requirements, several new technologies have been, and will continue to be, introduced in heavy-duty trucks. This leads to an increased complexity of the truck's engine and aftertreatment system and to an increased complexity of the supervisory control system that is used to exploit the synergy between the engine and aftertreatment system. Traditional supervisory control strategies are often based on heuristic rules, which do not lead to optimal solutions. It is believed that an increased fuel efficiency can be realised by designing an improved supervisory control strategy that uses ideas from optimal control theory. Such a strategy is sometimes referred to as Integrated Emission Management (IEM).

Some results on emission management can be found in the literature (Ao, Qiang, Zhong, Mao, & Yang, 2008; Serrao et al., 2013). In IEM, as proposed in Cloudt and Willems (2011) and experimentally demonstrated in Willems, Mentink, Kupper, and Van Den Eijnden (2013), the objective is to minimise operational costs, while satisfying emission legislation constraints. This is achieved by a delicate combination of doing Exhaust Gas Recirculation (EGR) in the engine, to reduce engine-out NO_x at the cost of higher fuel consumption, and converting NO_x in the engine's aftertreatment system, which leads to a higher AdBlue consumption. The IEM strategy of Cloudt and Willems (2011) and Willems et al. (2013) uses ideas from Energy Management Systems (EMS) for hybrid electric vehicles, which are extensively discussed in, e.g., Onori and Serrao (2011),Pisu and Rizzoni (2007), Sciarretta and Guzzella (2007), Salmasi (2007), Koot et al. (2005), and the book de Jager, van Keulen, and Kessels (2013). In the work on EMS, Dynamic Programming (DP) is often used to find the optimal solution. As this optimal solution is inherently noncausal and requires the drive cycle to be known a priori, suboptimal real-time implementable solutions have been proposed in the form of an Equivalent Cost Minimisation Strategy (ECMS). ECMS is based on

☆A preliminary version of this paper has been presented at the 2014 IFAC World Congress.

* Corresponding author.
E-mail address: m.c.f.donkers@tue.nl (M.C.F. Donkers).

http://dx.doi.org/10.1016/j.conengprac.2016.03.006
0967-0661/

Pontryagin's Minimum Principle (PMP) and aims at approximating the optimal solution. Although the real-time implementable solution to the IEM problem of Cloudt and Willems (2011) is based on PMP and has a resemblance to ECMS, a comparison of this real-time solution with the optimal solution obtained through DP has only been made in a preliminary version of this paper (Van Schijndel, Donkers, Willems, & Heemels, 2014). Still, Van Schijndel et al. (2014) consider a simplified dynamic model, and only use DP without making a comparison with the optimal solution obtained through the application of PMP. Comparing the optimal solution with the real-time implementable solution is relevant, as the experimental results of IEM presented in Willems et al. (2013) show a 2.1% fuel consumption reduction and a 1.5% total operational cost reduction with respect to a Euro-VI production type powertrain, which raises the question how much further the IEM results of Willems et al. (2013) can be improved.

This paper presents the optimal solution to the IEM problem for the diesel engine of a Euro VI heavy-duty truck completing a cold-start World Harmonised Transient Cycle (WHTC). A cold start is considered, since this cycle is more challenging from a thermal management and emissions point of view. The main contribution of this paper is a comparison of the real-time implementable strategy with the exact optimal solutions obtained through both DP and the application of PMP. The comparison between DP and PMP is relevant, as DP suffers from the 'curse of dimensionality', while PMP does not. However, PMP only provides a necessary condition for optimality, while DP is guaranteed to provide the global optimal solution.

A second contribution is an investigation, a comparison and the proposition of two novel extensions to existing DP algorithms. Namely, as emission legislation poses a constraint on the terminal state, the standard DP algorithm is sensitive to numerical errors (Sundström, Ambühl, & Guzzella, 2010). These numerical errors are caused by interpolating between grid points in general, and between finite and infinite costs in particular, where the latter are induced by the terminal state constraint. We will first review a baseline DP algorithm and the algorithm proposed by Elbert, Ebbesen, and Guzella (2013). Subsequently, we will extend this latter algorithm by adding a fast overapproximation of the so-called forward reachable sets. Namely, by approximating the forward reachable sets, the grid size over time can be reduced, which yields smaller numerical errors due to better interpolation. To reduce the numerical errors due to interpolation between finite and infinite cost, we approximate the backward reachable sets in the DP algorithm. This can be done using an approach based on an extension of Sundström et al. (2010). In particular, as the work of Sundström et al. (2010) can only be applied to scalar-state systems, we extend the aforecited work towards the particular higher-order system of IEM. The potential of the methods will be demonstrated by a simulation study of a cold-start WHTC. This provides insight into how the real-time implementable solution and optimal solutions to the IEM problem cope with the drive cycle and shows that the novel extension to the DP algorithm outperforms the existing DP algorithms.

The outline of this paper is as follows. First, a problem description is given in Section 2, including the objectives of this paper, the model of the engine plus aftertreatment system and the formulation of IEM as an optimal control problem. In Section 3, we show four different DP approaches and explain their advantages and disadvantages. This includes our novel extensions to existing DP algorithms. In Section 4, the optimal solution using PMP is discussed and the real-time implementable solution that which is based on PMP is discussed. The discussed DP and PMP methods are applied to the IEM problem and the optimal costs are compared in Section 5 and, finally, conclusions are drawn in Section 6.

2. Problem description

The objective of this paper is to find the optimal solution to the Integrated Emission Management (IEM) problem, which corresponds, loosely speaking, to *minimising engine operational costs* over a drive cycle while *satisfying emission constraints* imposed by legislation. Before being able to formalise the IEM problem, which we will do at the end of this section, we will first discuss the controller structure and present the necessary models.

2.1. System description and control structure

In this paper, we consider a state-of-the-art Euro VI heavy-duty powertrain, consisting of a 6 cylinder, 12.9 l, 375 kW engine and an Engine Aftertreatment System (EAS), see Fig. 1. The engine is equipped with a cooled high-pressure Exhaust Gas Recirculation (EGR) system and a Variable Turbine Geometry (VTG) with charge-air cooler. The EAS consists of a Diesel Oxidation Catalyst (DOC), a Diesel Particulate Filter (DPF), a 32.6 l Cu-Zeolite Selective Catalytic Reduction catalyst (SCR) and an Ammonia Oxidation catalyst (AMOX).

A block diagram of the control structure can be found in Fig. 2. The amount of fuel fed to the engine is taken so that the requested power is delivered. The AdBlue dosing strategy, which controls the SCR, aims at achieving maximal NO_x conversion. It is calibrated for a certain amount of NH_3-slip and is assumed to work autonomously. The air management controls the position of the EGR valve and the change of geometry of the VTG. The supervisory control strategy determines the desired EGR and VTG mass flows, which result in a tradeoff between emissions and fuel economy. As a first step towards the optimisation of the operational costs, we focus on NO_x emissions, because present Euro VI technology is capable of reducing other emissions such that they comply with the Euro VI legislation. DPF regeneration is outside the scope of this paper.

2.2. Engine and aftertreatment models

The EAS is described by a dynamic model, whereas the engine is described by a static (stationary) model. This is because the physical phenomena occurring in the EAS are relatively slow compared to the ones occurring in the engine. Both models have been parameterised and experimentally validated in Willems et al. (2013). We first discuss the static model of the engine, and subsequently the dynamic model of the EAS.

The engine model used in this paper is static (stationary) and is based on Wahlström and Eriksson (2011). In fact, it can be derived from Wahlström and Eriksson (2011) by assuming that all dynamics are in steady state. The inputs and outputs of the engine model are shown in Fig. 3. The model predicts the fuel mass flow $\dot{m}_f$ (kg s^{-1}), the total exhaust gas mass flow $\dot{m}_{\text{exh}}$ (kg s^{-1}), the engine-out NO_x mass flow $\dot{m}_{\text{NO}_x,\text{eo}}$ (kg s^{-1}) and the exhaust gas temperature T_{exh} (K) as a function of desired torque τ (N m) for a specific desired rotation speed ω (rad s^{-1}), and as a function of the mass flow through the EGR and VTG, denoted by u_1 (kg s^{-1}) and

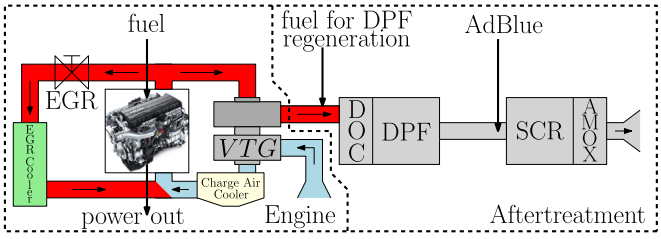


Fig. 1. Schematic overview of the powertrain.

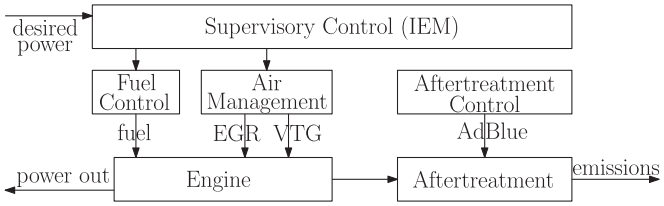


Fig. 2. Block diagram of the powertrain.

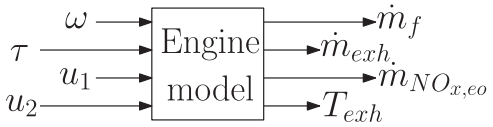


Fig. 3. Inputs and outputs of the static engine model.

u_2 (kg s^{-1}), respectively. The supervisory controller determines u_1 (kg s^{-1}) and u_2 (kg s^{-1}), where the allowable u_1 and u_2 depends on the combination of τ and ω, i.e, $(u_1, u_2) \in \tilde{\mathcal{U}}(\tau, \omega)$. The sets $\tilde{\mathcal{U}}(\tau, \omega)$ have been determined experimentally for a range of τ and ω by changing the setpoints for exhaust manifold pressure and engine-out NO_x of the air management and measuring the range of achieved u_1 and u_2. It is assumed that the air management and the fuel control perfectly track their setpoints given by the IEM strategy. The experimental results of Willems et al. (2013) suggest that this is a valid assumption. Finally, it should be mentioned that this model does not include engine warmup and we will reflect on this simplification in Section 5.

The EAS model is taken from Willems et al. (2013) and is described by the dynamic model

$$\begin{cases} \dot{T}_{DOC} &= \dfrac{c_{exh}}{C_{DOC}} \dot{m}_{exh}(T_{exh} - T_{DOC}), \\ \dot{T}_{SCR} &= \dfrac{c_{exh}}{C_{SCR}} \dot{m}_{exh}(T_{DOC} - T_{SCR}) + \dfrac{h_{amb}}{C_{SCR}}(T_{amb} - T_{SCR}), \\ \dot{m}_{NO_x,tp} &= \dot{m}_{NO_x,eo}(1 - \eta_{SCR}(T_{DOC}, T_{SCR}, \dot{m}_{exh})), \end{cases} \tag{1}$$

where T_{DOC} (K), T_{SCR} (K), T_{exh} (K) and T_{amb} (K) are the temperature of the DOC, SCR, engine-out exhaust gas and the ambient, respectively, $\dot{m}_{exh}$ (kg s^{-1}) is the total mass flow of the engine-out exhaust gas, and $m_{NO_x,tp}$ (kg) is the cumulative amount of tailpipe NO_x emissions. The SCR efficiency η_{SCR} (–) is a nonlinear function that depends on the SCR temperature T_{SCR}, the DOC temperature T_{DOC} and the exhaust gas mass flow $\dot{m}_{exh}$. The SCR efficiency depends on DOC temperature, because the DOC temperature influences the NO_2/ NO_x-ratio in the exhaust gas, which affects SCR efficiency. The parameters c_{exh}, C_{SCR}, C_{DOC}, and h_{amb} are related to the heat transfer coefficients of the EAS and given in Table 1, and are determined experimentally.

2.3. Optimal control problem formulation

After presenting the required models, we can now formulate the IEM problem in the form of an optimal control problem. In this optimal control problem, the objective is to minimise the total fluid costs, while staying within the constraints on NO_x emissions.

Table 1
Parameters aftertreatment model.

Parameter name	Symbol	Unit	Value
Specific heat capacity exhaust gas	c_{exh}	J K^{-1} kg^{-1}	1000
Total heat capacity DOC	C_{DOC}	J K^{-1}	8600
Total heat capacity SCR	C_{SCR}	J K^{-1}	19 500
Ambient heat transfer coefficient	h_{amb}	J K^{-1} s^{-1}	15

More formally, this problem can be expressed as

$$J_c(x(t_s)) = \min_{(u_1,u_2)\in\mathcal{U}(t)} \int_{t_s}^{t_f} \pi_f \dot{m}_f(u, t) + \pi_A \dot{m}_A(T_{DOC}, T_{SCR}, u, t)\, dt, \tag{2}$$

with $\mathcal{U}(t) = \tilde{\mathcal{U}}(\tau(t), \omega(t))$, subject to (1) and

$$\underline{m}_{NO_x,tp} \le m_{NO_x,tp}(t_f) \le \overline{m}_{NO_x,tp}, \tag{3}$$

where $\pi_f = 1.34$ euro/kg and $\pi_A = 0.50$ euro/kg are the price of fuel and AdBlue, respectively, $\dot{m}_f$ is the fuel mass flow of the engine model, $\underline{m}_{NO_x,tp}$ (kg) and $\overline{m}_{NO_x,tp}$ (kg) are the terminal state constraints on NO_x emissions, which represents the minimum and maximum amount of NO_x that is allowed to be emitted over the drive cycle used for type approval, respectively, where, typically, $\underline{m}_{NO_x,tp} = 0$ kg, and t_s (s) and t_f (s) are the initial and terminal time of the cycle, respectively.

While the fuel mass flow $\dot{m}_f$ in (2) comes from the engine model, see Fig. 3, the AdBlue dosing $\dot{m}_A$ in (2) is a result from the assumption that the reaction with NO_x is stoichiometric, all AdBlue are used for NO_x conversion, and that the aftertreatment control aims at maximising this NO_x conversion. As such, the AdBlue mass flow is given by

$$\dot{m}_A = \nu\, \dot{m}_{NO_x,eo}\, \eta_{SCR}(T_{DOC}, T_{SCR}, \dot{m}_{exh}), \tag{4}$$

where $\nu = 2.007$ is the stoichiometric ratio between AdBlue and NO_x, and $\dot{m}_{NO_x,eo}$ and η_{SCR} are given as before.

2.4. Model-order reduction and discrete-time approximation

For the solution methods presented below, the third-order model (1) is reduced to a second-order model and, subsequently, approximated in discrete-time to make it amenable for Dynamic Programming (DP). The main reason for reducing the model order is that this simplifies the computations, particularly for DP, while it still allows for a fair comparison between different methods. Note that the methods presented below will also be applied to the third-order model in the simulation study, as well, even though the higher complexity restricts the accuracy of the solutions of DP.

To obtain the second-order model, we describe the temperatures T_{DOC} and T_{SCR} by one single temperature T_{EAS} that represents both catalytic converters. To achieve this, the heat capacity of the entire system is taken as the sum of the heat capacities of the individual components, resulting in

$$\dot{x} = f(x, u, t) := \begin{bmatrix} k_1 \dot{m}_{exh}(T_{exh} - x_1) + k_2(T_{amb} - x_1) \\ \dot{m}_{NO_x,eo}(1 - \eta_{SCR}(x_1, \dot{m}_{exh})) \end{bmatrix}, \tag{5}$$

where $x = [x_1\ x_2]^T = [T_{EAS}\ m_{NO_x,tp}]^T$. The parameters $k_1 = c_{exh}/(C_{DOC} + C_{SCR})$ and $k_2 = h_{amb}/(C_{DOC} + C_{SCR})$ result from lumping the heat capacity of the two catalytic converters. As a consequence of this simplification, the optimal control problem for this reduced-order model becomes (2) and (4), in which T_{DOC} and T_{SCR} is replaced by $T_{EAS} = x_1$, subject to (5) and (3).

As a last step before being able to apply DP and PMP, the optimal control problem given by (2) subject to (5) and (3) is approximated in discrete time. To do so, we apply the forward Euler method with step size of 1 s, resulting in

$$J(x[0]) = \min_{\{u[k]\}_{k=0}^{N-1}\in\mathcal{U}^t} \sum_{k=0}^{N-1} G(x[k], u[k], k) + G(x[N], N), \tag{6}$$

subject to

$$x[k+1] = F(x[k], u[k], k) := x[k] + f(x[k], u[k], k), \tag{7}$$

with $f(x, u, k)$ as in (5), $x[k] = x(t_s + k)$, $u[k] = u(t_s + k)$, and

$$L(x[N]) \le 0. \tag{8}$$

We use the forward Euler method and a step size of 1 s because of the simplicity of the method and because of the fact that the dynamics are associated with slow thermal effects (significantly slower than 1 s). In (6), the running cost is given by

$$G(x, u, k) = \pi_f \dot{m}_f(u, k) + \pi_A \dot{m}_A(x_1, x_1, u, k), \tag{9}$$

for $k = \{0, 1, \ldots, N-1\}$, the terminal cost by $G(x, N) = 0$, the terminal constraint by

$$L(x) = \begin{bmatrix} x_2 - \overline{m}_{\mathrm{NO}_x,\mathrm{tp}} \\ \underline{m}_{\mathrm{NO}_x,\mathrm{tp}} - x_2 \end{bmatrix}, \tag{10}$$

and $\mathcal{U}^t = \mathcal{U}[0] \times \ldots \times \mathcal{U}[N-1]$.

3. Dynamic programming

We will now solve the discrete-time optimal control problem (6)–(8) using Dynamic Programming (DP), see, e.g., Bertsekas (2005). DP requires solving the backwards recursion

$$J(x, k) = \min_{u \in \mathcal{U}[k]} \{G(x, u, k) + J(F(x, u, k), k+1)\}, \tag{11}$$

for all $k \in \{0, 1, \ldots, N-1\}$, subject to $J(x, N) = G(x, N)$ for all $x \in \mathcal{X}^b[N]$ and $J(x, N) = \infty$ for all $x \notin \mathcal{X}^b[N]$, where

$$\mathcal{X}^b[N] = \{x \in \mathbb{R}^n | L(x) \leq 0\}. \tag{12}$$

The function $J(x, k)$ is often called the cost-to-go function. The corresponding optimal inputs u^* are defined as

$$u^*(x, k) \in \arg\min_{u \in \mathcal{U}[k]} \{G(x, u, k) + J(F(x, u, k), k+1)\}. \tag{13}$$

We will now discuss four different methods to solve the recursion (11). For each of these methods, we perform a spatial discretisation of a relevant parts of the state space $\mathcal{X}[k] \subset \mathbb{R}^n$ and the input space $\mathcal{U}[k]$, $k \in \{0, 1, \ldots, N-1\}$, which results in countable sets $\mathcal{X}^o[k] \subset \mathcal{X}[k]$, $\mathcal{U}^o[k] \subset \mathcal{U}[k]$. We use linear interpolation and extrapolation to obtain values for $x[k] \in \mathcal{X}[k]$ that are not contained in $\mathcal{X}^o[k]$. The accuracy of the solution of the DP problem can be increased by increasing the number of elements in $\mathcal{X}^o[k]$ and $\mathcal{U}^o[k]$ in a well-distributed way, albeit at the cost of a higher computational complexity. Although the choice of $\mathcal{U}^o[k]$ affects the accuracy of the solution, we will not focus this choice in this paper. The difference between the DP methods discussed below is how to choose the set $\mathcal{X}^o[k]$, $k \in \{0, 1, \ldots, N\}$.

3.1. Basic dynamic programming

In the most basic implementation of dynamic programming, denoted as Basic DP (BDP), the terminal state constraint (8) is taken into account by making the terminal cost of the infeasible area, i.e., the region where $L(x(N)) > 0$, large, see, e.g., Sundström et al. (2010) and Elbert et al. (2013). This means that we can solve the DP problem by initialising

$$J(x, N) = \begin{cases} G(x, N), & \text{if } L(x) \leq 0, \\ \alpha, & \text{if } L(x) > 0, \end{cases} \tag{14}$$

for $x \in \mathcal{X}^o[N]$, where $\alpha \in \mathbb{R}$ is chosen sufficiently large. This causes the entire state space to become feasible and allows us to select a time-invariant and countable set $\mathcal{X}^I$ that has its points well distributed over a sufficiently large part of the state space $\mathbb{R}^n$, leading to $\mathcal{X}^o[k] := \mathcal{X}^I$ for all $k \in \{0, 1, \ldots, N-1\}$. Then, for all $x \in \mathcal{X}^o[k]$, we compute

$$J(x, k) = \min_{u \in \mathcal{U}^o[k]} \{G(x, u, k) + J(F(x, u, k), k+1)\} \tag{15}$$

for k ranging from $N-1$ back to 0, subject to (14).

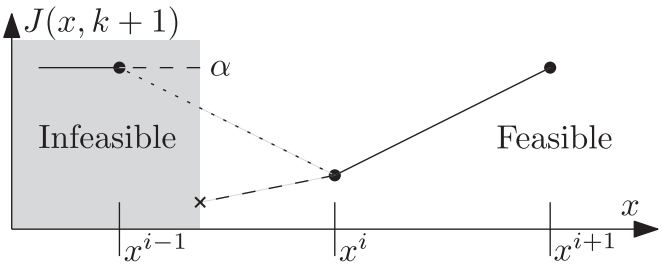


Fig. 4. Possible interpolation error at the boundary in 1D.

BDP is the most straightforward way to solve a DP problem with terminal state constraints. However, the fact that we grid and approximate infeasible points in the state space by a large but finite number α causes the algorithm to suffer from numerical errors due to interpolation at the boundary (Sundström et al., 2010). As illustrated in Fig. 4, the true minimal cost is achieved somewhere in between grid points x^{i-1} and x^i, but due to interpolation, the minimum appears to occur at x^i. In this figure, the dotted line illustrates the interpolation when doing basic DP, while the dashed line illustrates the real cost-to-go and could be captured by a grid point (cross) on the boundary of the feasible area. To minimise the numerical errors using BDP, we have to make sure that α is not too high, yet high enough to avoid it from affecting the solution to (15), because of the interpolation errors indicated in Fig. 4.

As was also concluded in Sundström et al. (2010), the aforementioned interpolation errors are undesired and we will discuss an existing solution for this issue. Moreover, we propose two different extensions to existing DP algorithms to solve this problem, as well.

3.2. Level-set dynamic programming

An existing solution to the numerical errors due to interpolation is called Level-Set Dynamic Programming (LSDP), which is described by Elbert et al. (2013). It keeps track of the so-called backward reachable sets, which are, loosely speaking, the sets of $x[k]$, $k \in \{0, 1, \ldots, N-1\}$, for which there exist a solution to (7) that satisfies $x[N] \in \mathcal{X}^b[N]$, with $\mathcal{X}^b[N]$ as in (12). Keeping track of the backwards reachable sets can be done using a function $\mathcal{I}(x, k)$ that satisfies $\mathcal{I}(x, k) \leq 0$ if x is backward reachable at time step k and $\mathcal{I}(x, k) > 0$ if it is not. The function $\mathcal{I}(x, k)$ satisfies the following recursion

$$\mathcal{I}(x, k) = \min_{u \in \mathcal{U}^o[k]} \{\mathcal{I}(F(x, u, k), k+1)\}. \tag{16}$$

for k ranging from $N-1$ back to 0, for all $x \in \mathcal{X}^o[k]$, subject to $\mathcal{I}(x, N) = L(x)$.

Instead of interpolating between the value of a point within the backwards reachable set and a large but finite value α for x not in the backward reachable set, the LSDP method tries to make a smooth transition between the optimal costs of feasible and infeasible points. This is done by using an estimate of the costs for infeasible states based on the inputs that steer the dynamics "closest" to the feasible set. This is expressed by only allowing control inputs $u \in \mathcal{U}^F(x, k)$ to be used, where

$$\mathcal{U}^F(x, k) = \mathcal{U}^{F_1}(x, k) \bigcup \mathcal{U}^{F_2}(x, k) \tag{17}$$

for $x \in \mathcal{X}^o[k]$, $k \in \{0, 1, \ldots, N-1\}$, with

$$\mathcal{U}^{F_1}(x, k) = \{u \in \mathcal{U}^o[k] | \mathcal{I}(F(x, u, k), k+1) \leq 0\} \tag{18a}$$

$$\mathcal{U}^{F_2}(x, k) = \arg\min_{u \in \mathcal{U}^o[k]} \{\mathcal{I}(F(x, u, k), k+1)\}. \tag{18b}$$

The set $\mathcal{U}^{F_1}(x, k)$ is the set of control inputs that allow $x \in \mathcal{X}^o[k]$ to be backwards reachable (and can be the empty set for some $x \in \mathcal{X}^o[k]$), while $\mathcal{U}^{F_2}(x, k)$ is the set of control inputs that steer the dynamics "closest" to the backwards reachable set (and can be a singleton). The latter set satisfies $\mathcal{U}^{F_2}(x, k) \subset \mathcal{U}^{F_1}(x, k)$ in case $\mathcal{U}^{F_1}(x, k) \neq \varnothing$. The set $\mathcal{U}^F(x, k)$ can now be used to calculate the cost-to-go

$$J(x, k) = \min_{u \in \mathcal{U}^F(x,k)} \{G(x, u, k) + J(F(x, u, k), k + 1)\} \tag{19}$$

for k ranging from $N - 1$ back to 0, $x \in \mathcal{X}^o[k]$, subject to $J(x, N) = G(x, N)$. Similar to BDP, the time-invariant set $\mathcal{X}^I$ is used, so that $\mathcal{X}^o[k] := \mathcal{X}^I$ for all $k \in \{0, 1, \ldots, N - 1\}$.

Although LSDP implicitly deals with the backward reachable sets, the so-called forward reachable sets, which are defined as the sets of states $x[k]$, $k \in \{1, 2, \ldots, N\}$, which can be reached from the set of initial conditions, are not taken into account at all. This means that the cost-to-go is computed for points that can never be reached, making the computational complexity unnecessarily high or equivalently, the accuracy for a given computational complexity too small.

3.3. Extended level-set dynamic programming

A first contribution of this paper is called Extended LSDP (ELSDP), which is a slight extension of the LSDP method. The idea is to include an overapproximation of the forward reachable sets in the LSDP approach, which will make the set $\mathcal{X}[k] \subset \mathbb{R}^n$ time-varying and prevent $\mathcal{X}[k]$ from being unnecessary large for $k \in \{0, 1, \ldots, N\}$.

We will include an approximation $\mathcal{X}^f[k]$ of the forward reachable sets as follows. For a given $\mathcal{X}^f[0]$, we first find the extreme values of the components x_i of x in the set $\mathcal{X}^f[k]$ by

$$\underline{x}_i[k] = \min\{x_i \in \mathbb{R} | x \in \mathcal{X}^f[k]\}, \tag{20a}$$

$$\overline{x}_i[k] = \max\{x_i \in \mathbb{R} | x \in \mathcal{X}^f[k]\}, \tag{20b}$$

for $i \in \{1, \ldots, n\}$, where n is the dimension of the state-space, i.e., $\mathcal{X}^f[k] \subset \mathbb{R}^n$. This leads to an overapproximation of the set $\mathcal{X}^f[k]$ by the hyperrectangle with the vertices

$$\hat{\mathcal{X}}^f[k] := \{x \in \mathbb{R}^n | x_i \in \{\underline{x}_i[k], \overline{x}_i[k]\}, i \in \{1, \ldots, n\}\}. \tag{21}$$

Finally, we compute

$$\mathcal{X}^f[k + 1] := \text{co}\{F(x, u, k) \in \mathbb{R}^n | x \in \hat{\mathcal{X}}^f[k], u \in \mathcal{U}^o[k]\}. \tag{22}$$

using the set of vertices $\hat{\mathcal{X}}^f[k]$ of the hyperrectangle, where the notation co $\mathcal{X}$ denotes the convex hull of a set $\mathcal{X}$. We then take the set $\mathcal{X}[k] = \{x \in \mathbb{R}^n | \underline{x}_i[k] \leq x_i \leq \overline{x}_i[k], i \in \{1, \ldots, n\}\}$ for all $k \in \{0, 1, \ldots, N - 1\}$, take a set of grid point $\mathcal{X}^o[k] \subset \mathcal{X}[k]$ and use the LSDP algorithm as explained above. Note that a requirement for this extension is that the extreme values of $\mathcal{X}^f[k + 1]$ resulting from (22) should be equal to the extreme values of the set $\{F(x, u, k) \in \mathbb{R}^n | x \in \mathcal{X}^f[k], u \in \mathcal{U}^o[k]\}$.

By using an overapproximation of the forward reachable sets, we try not to place grid points $x \in \mathcal{X}^o[k] \subset \mathcal{X}[k]$, $k \in \{0, 1, \ldots, N - 1\}$, at positions that can never be reached from $\mathcal{X}^f[0]$. This means that (the same number of) grid points can be closer to each other, yielding a more accurate solution of LSDP with the same computational complexity. However, neither LSDP nor ELDSP methods solve the real issue of interpolation errors near the boundary of the backward reachable sets (i.e., the feasible regions), as was shown in Fig. 4. To do so, we will propose an extension of Sundström et al. (2010) towards higher-order systems.

3.4. Boundary-surface dynamic programming

The second contribution of this paper is Boundary-Surface Dynamic Programming (BSDP), which can be seen as an extension to Boundary-Line Dynamic Programming (BLDP) (Sundström et al., 2010). The extension is needed as BLDP as in Sundström et al. (2010) can only be applied to scalar-state systems. The main idea of BSDP, as well as BLDP of Sundström et al. (2010), is to include grid points that are on the boundary of the backward reachable sets to avoid the interpolation errors illustrated in Fig. 4. In the BSDP method that we propose, we also compute an approximation of the forward reachable sets $\mathcal{X}^f[k]$. Computations of these two sets will, in principle, allow us to place grid points $x \in \mathcal{X}^o[k]$, $k \in \{0, 1, \ldots, N\}$, exactly on the boundary of the intersection of the forward and backward reachable sets, thereby having no grid points x outside the reachable sets. Hence, the set $\mathcal{X}^o[k]$, $k \in \{0, 1, \ldots, N\}$, consists of grid points that lie within the intersection of the approximated forward and backward reachable sets for the BSDP algorithm.

To compute the forward and backward reachable sets, we propose an algorithm that takes into account the specific structure of the IEM problem in $\mathbb{R}^2$, i.e., $n=2$, of Section 2.4. We assume that the dynamics (7) are boundary preserving in the sense that the boundary of a set that is mapped using the dynamics (7) is equal to the mapping of the boundary of that same set. This property is guaranteed under certain homeomorphism conditions on the mapping F as in (7) and we use this property to our advantage by keeping track of the boundary instead of the complete reachable sets.

To compute approximations $\mathcal{X}^f[k]$, $k \in \{0, 1, \ldots, N\}$, of the forward reachable sets for a given convex set of initial conditions $\mathcal{X}^f[0] \subset \mathbb{R}^2$, the boundary of the forward reachable set is approximated by computing

$$\underline{x}_1[k] = \min\{x_1 \in \mathbb{R} | x \in \mathcal{X}^f[k]\}, \tag{23a}$$

$$\overline{x}_1[k] = \max\{x_1 \in \mathbb{R} | x \in \mathcal{X}^f[k]\}, \tag{23b}$$

and defining

$$\tilde{x}_1^i[k] = \frac{i - 1}{s - 1} \underline{x}_1[k] + \frac{s - i}{s - 1} \overline{x}_1[k], \tag{24}$$

for $i \in \{1, \ldots, s\}$, for some $s \in \mathbb{N}$, allowing us to compute

$$\underline{x}_2^i[k] = \min\left\{x_2 \in \mathbb{R} \left| \begin{bmatrix} \tilde{x}_1^i[k] \\ x_2 \end{bmatrix} \in \mathcal{X}^f[k] \right.\right\}, \tag{25a}$$

$$\overline{x}_2^i[k] = \max\left\{x_2 \in \mathbb{R} \left| \begin{bmatrix} \tilde{x}_1^i[k] \\ x_2 \end{bmatrix} \in \mathcal{X}^f[k] \right.\right\}, \tag{25b}$$

to obtain the following approximation of the boundary of the forward reachable set

$$\hat{\mathcal{X}}^f[k] := \bigcup_{i=1}^{s} \left\{ \begin{bmatrix} \tilde{x}_1^i[k] \\ \underline{x}_2^i[k] \end{bmatrix}, \begin{bmatrix} \tilde{x}_1^i[k] \\ \overline{x}_2^i[k] \end{bmatrix} \right\}. \tag{26}$$

The final step is to compute

$$\mathcal{X}^f[k + 1] := \text{co}\{F(x, u, k) \in \mathbb{R}^n | x \in \hat{\mathcal{X}}^f[k], u \in \mathcal{U}^o[k]\}. \tag{27}$$

Indeed, (27) computes an approximation of the forward reachable set as it maps all the values x can attain at time k to all the values the state x can attain at time $k + 1$. Note that the approximation $\mathcal{X}^f[k]$ of the forward reachable set is found in a simple way by using the convex hull operation in (27) and that by increasing

$s \in \mathbb{N}$, we are able to obtain a more accurate approximation of the true forward reachable set. Note also that by taking the convex hull we will typically create an overapproximation of the forward reachable set, meaning that we capture at least all forward reachable points, but also might include some points that are not forward reachable. However, the convex hull operation significantly reduces the computational complexity and, as it turns out, will lead, for the system under study, to a good approximation of the true forward reachable sets. To find all forward reachable sets, we solve (23)–(27) recursively for $k \in \{0, 1, \ldots, N-1\}$.

To compute the backward reachable sets, we discuss an extension to Sundström et al. (2010) towards a second-order system with a special structure. In particular, we require that the terminal state constraint $\mathcal{X}^b[N]$ given by the mapping L as in (10) can be written as $\mathcal{X}^b[N] = \{x \in \mathbb{R}^2 | \underline{r}(x_1) \le x_2 \le \overline{r}(x_1)\}$ for some functions $\underline{r}, \overline{r}: \mathbb{R} \to \mathbb{R}$. To avoid unnecessary computations, we directly incorporate the overapproximation $\mathcal{X}^f[k]$ of the forward reachable sets. Therefore we use (24), so that for a given $\mathcal{X}^b[N]$, the boundary of the backward reachable set is recursively approximated by computing for $i \in \{1, \ldots, s\}$

$$\underline{x}_2^i[k] = \min_{u \in \mathcal{U}^o[k]} \{x_2 \in \mathbb{R} \left| F\left(\begin{bmatrix} \tilde{x}_1^i[k] \\ x_2 \end{bmatrix}, u, k\right) \in \tilde{\mathcal{X}}^b[k+1], \begin{bmatrix} \tilde{x}_1^i[k] \\ x_2 \end{bmatrix} \in \mathcal{X}^f[k] \right\}. \tag{28a}$$

$$\overline{x}_2^i[k] = \max_{u \in \mathcal{U}^o[k]} \{x_2 \in \mathbb{R} \left| F\left(\begin{bmatrix} \tilde{x}_1^i[k] \\ x_2 \end{bmatrix}, u, k\right) \in \tilde{\mathcal{X}}^b[k+1], \begin{bmatrix} \tilde{x}_1^i[k] \\ x_2 \end{bmatrix} \in \mathcal{X}^f[k] \right\}. \tag{28b}$$

The approximation of the backward reachable set is then computed with

$$\begin{aligned} \mathcal{X}^b[k] := & \bigcup_{i=1}^{s-1} \mathrm{co}(\{\tilde{x}_1^i[k]\} \times [\underline{x}_2^i[k], \overline{x}_2^i[k]] \\ & \bigcup \{\tilde{x}_1^{i+1}[k]\} \times [\underline{x}_2^{i+1}[k], \overline{x}_2^{i+1}[k]]). \end{aligned} \tag{29}$$

To find all backward reachable sets, we solve (28) and (29) recursively for $k \in \{0, 1, \ldots, N-1\}$.

The approximations of the forward and backward reachable sets can be used to place all grid points on the boundary and in the interior of the intersection of these two sets, i.e., $x \in \mathcal{X}^o[k] \subset (\mathcal{X}^f[k] \cap \mathcal{X}^b[k])$ for all $k \in \{0, 1, \ldots, N\}$. For our system, we can find these sets with relatively low computational effort, which gives increased accuracy for only a slightly increased computational time. Moreover, it provides insight into the solution, as it allows us to analyse the optimal trajectory with respect to the reachable sets, which is a result that does not follow from the other methods.

The computation of the forward and backward reachable sets for the third-order model (1) are not discussed in this paper. Still, the procedure presented above can be applied to the third-order model *mutatis mutandis*.

4. Pontryagin's minimum principle

Instead of using DP, the optimal control problem (6)–(8) can also be solved by applying Pontryagin's Minimum Principle (PMP). PMP says that the optimal solution $u^*[k]$ along an optimal trajectory $x^*[k]$ satisfies

$$u^*[k] \in \arg\min_{u \in \mathcal{U}^o[k]} H(x^*[k], u, k), \tag{30a}$$

$$\lambda[k] = \frac{\partial H(x^*[k], u^*[k], k)}{\partial x^*[k]}, \tag{30b}$$

for $k \in \{0, 1, \ldots, N-1\}$, subject to (7) and the transversality condition $\lambda^\top[N]\,\delta x[N] = 0$, see, e.g., Bertsekas (2005). The transversality condition states that whenever (some) $x_i[N]$ are given (i.e., $\delta x_i[N] = 0$), the corresponding $\lambda_i^\top[N]$ can have any value, and that whenever (some) $x_i[N]$ are not given, it has to hold that $\lambda_i^\top[N] = 0$. In (30), the Hamiltonian is given by

$$H(x, u, k) = G(x, u, k) + \lambda^\top[k+1] F(x, u, k), \tag{31}$$

where $G(x, u, k)$ is the running cost as in (6), $\lambda[k]$ are the so-called costates, $k \in \{0, 1, \ldots, N\}$, and $F(x, u, k)$ are the state dynamics as in (7). It should be noted that PMP provides a necessary condition for optimality, meaning that the optimal solution satisfies (30), but there might also exist solutions to (30) that are not optimal. Below, we propose a numerical method that aims at solving (30) and a method that aims at approximating (30), such that the resulting solution can be implemented in real time.

4.1. Optimal solution using PMP

We will first present a method to solve the optimal control problem by applying PMP, which we will denote by Optimal PMP (O-PMP). To do so, we substitute (7), with (5), and (9) into (31) to obtain the dynamics for the costates (30b) as

$$\begin{cases} \lambda_1[k] = \lambda_1[k+1] - h\lambda_1[k+1](k_1 \dot{m}_{\mathrm{exh}}(u[k], k) + k_2)(\nu\pi a - h\lambda_2[k+1]) \\ \qquad \times \dot{m}_{NO_{x,eo}}(u[k], k) \frac{\partial \eta_{\mathrm{SCR}}(x_1[k], u[k])}{\partial x_1[k]}, \\ \lambda_2[k] = \lambda_2[k+1]. \end{cases} \tag{32}$$

We observe that λ_2 is constant over the complete cycle, and that λ_1 is anticausal and that it is subject to $\lambda_1[N] = 0$. Because $x[0]$, $x_2[N]$ and $\lambda_1[N]$ are given, obtaining the (candidate) optimal solution using PMP involves solving a two-point boundary-value problem. To solve this two-point boundary value problem, the forward-backward sweep method from McAsey, Mou, and Han (2012) is adapted and used to find $\lambda_1[k]$ in (32) for all $k \in \{0, 1, \ldots, N\}$. From experience, we find that the minimal cost satisfying the terminal state constraint is obtained at $L(x(N)) = 0$ with $\underline{m}_{\mathrm{NO_x,tp}} = \overline{m}_{\mathrm{NO_x,tp}}$. This leads to the following procedure to find a constant value for λ_2:

1. Initialise $\lambda_1[k] = 0$ and $\lambda_2[k] = \tilde{\lambda}_2$ for all $k \in \{0, 1, \ldots, N\}$ and for some $\tilde{\lambda}_2 > 0$.
2. Solve (7) with (30a) from $k=0$ to $N-1$, for fixed $\lambda[k]$ and find values for $x[k]$, $\dot{m}_{\mathrm{exh}}(u[k], k)$, $\dot{m}_{NO_{x,eo}}(u[k], k)$, $\frac{\partial \eta_{\mathrm{SCR}}(x_1[k], u[k])}{\partial x_1[k]}$. Use these values to solve (32) from $k = N-1$ to 0. Repeat this step until $x[k]$ and $\lambda[k]$ have not changed significantly from the previous iteration to the current.
3. If $\| L(x(N)) \| < tol$, with $L(x)$ as in (8), then terminate. If $x_2[N] > \overline{m}_{\mathrm{NO_x,tp}}$, increase $\tilde{\lambda}_2$, or if $x_2[N] < \underline{m}_{\mathrm{NO_x,tp}}$, decrease $\tilde{\lambda}_2$ and go to step 1 of the procedure.

Note that this algorithm is not guaranteed to converge and, when it does, the solution is not necessarily the optimal solution. Therefore, the solution will be compared with the solution obtained through DP.

4.2. Real-time solution based on PMP

Besides the optimal solution to the control problem, we will also propose a heuristic real-time implementable suboptimal solution, which we will denote by RT-PMP and has resemblance to ECMS. This solution is based on the observation in Willems et al. (2013) that λ_1 has little influence on the minimal cost over a drive cycle. We therefore propose to simplify the problem by setting $\lambda_1[k] = 0$ and $\lambda_2[k] = \tilde{\lambda}_2$ for some $\tilde{\lambda}_2 \in \mathbb{R}$ and for all $k \in \{0, 1, \ldots, N\}$, where the constant $\tilde{\lambda}_2$ is selected such that the operational costs

are minimal over the cycle while satisfying (8). This allows the IEM problem to be solved using a simple shooting method to search for $\tilde{\lambda}_2$, or by simply taking a fixed value that is tuned over a representative cycle, as was done in Willems et al. (2013).

5. Simulation study

We will now demonstrate and compare the potential of the proposed solution strategies to the optimal control problem using a case study of a typical type approval drive cycle. First, we discuss the implementation details of the methods discussed in this paper. Next, we compare the accuracy in terms of obtained cost of the solution to the IEM problem using the DP and the PMP approaches for the second-order dynamic model of the EAS (5). Finally, we will discuss the trajectories corresponding to the different solutions to assess the performance of the newly proposed DP method, as well as the amount of performance degradation of the real-time implementable solution. The latter will be done for both the second-order EAS model (5), as well as the third-order EAS model (1).

In this simulation study, the World Harmonised Transient Cycle (WHTC) will be used, which is a 30-min (1800 s) routine for a heavy-duty truck translated to a transient engine dynamometer cycle. It is based on typical routes through urban areas (0–900 s), rural areas (900–1380 s) and highway areas (1380–1800 s) that heavy-duty vehicles drive in the real world. In a complete powertrain approval process, the WHTC is run twice: First with a cold engine and after a soak time of 10 min with a warm engine. Since the cold-start case is more challenging from a thermal management and emissions point of view, we will consider only the cold-start case, which corresponds to $T_{\mathrm{SCR}}[0] = T_{\mathrm{DOC}}[0] = T_{\mathrm{EAS}}[0] = 298$ K and $T_{\mathrm{amb}} = 298$ K. Note that NO_x prediction of the engine model is not realistic, as engine warmup has been neglected. Still, the cold-start case and the models employed in this paper allow for demonstrating NO_x reduction through combined air management (EGR/VTG) and thermal management of the EAS, which warms up much slower than the engine.

For a full powertrain approval process, the maximum amount of allowed tailpipe NO_x emissions is a weighted average over the cold-start and warm-start cycles. From Cloudt and Willems (2011), we find that an emission limit of 0.8 g/kWh is a good choice for the cold-start cycle, so that we are still able to achieve the legislation limit for the full type approval procedure.

5.1. Implementation details

Before showing the results of the simulation study, let us discuss some of the implementation details for each of the applied methods. For all methods, we take $q = 21^2$ equidistantly spaced points for all $k \in \{0, 1, \ldots, N-1\}$ for the number of grid points of $\mathcal{U}^o[k] \subset \mathcal{U}[k]$. The forward–backward sweep method used to find the optimal solution using PMP, as discussed in Section 4, is solved in 10 iterations and using a tolerance $tol = 0.15 \cdot 10^{-3}$ kg.

For BDP and LSDP, we have to specify grid points $\mathcal{X}^I$. We choose the grid points $\{x^i\}_{i=1}^p = \mathcal{X}^I$ to be on a regularly-spaced rectangular grid with an equal number of quantisation levels for all states and we will vary the number of grid points $p \in \mathbb{N}$ to obtain more accurate solutions. We select 298 K and 900 K as lower and upper bounds for the temperatures, respectively, because these temperatures will never drop below the ambient temperature, and because larger temperatures are not forward reachable. For the cumulative tailpipe NO_x state $m_{\mathrm{NO}_x,\mathrm{tp}}$, we select 0 kg and 28.7×10^{-3} kg as lower and upper bounds, respectively, as the cumulative tailpipe NO_x cannot decrease, and the upper bound corresponds to the emission constraint of 0.8 g/kWh (as the requested power of the WHTC is known). The parameter α as in (14) is chosen based on maximising the operational costs in (2), subject to no emission constraints, which yields $\alpha = 11.67$.

For the ELSDP and BSDP solution strategy, we will also take regularly-spaced grid points $\{x^i\}_{i=1}^p = \mathcal{X}^o[k] \subset \mathcal{X}[k]$, $k \in \{0, 1, \ldots, N\}$, but the choice of grid points is different for each k. For ELSDP, we take a time-varying rectangular grid as was described in Section 3.3, and for BSDP, we use the method as outlined in Section 3.4, with $s = p^{1/n}$ for some $p \in \mathbb{N}$, with n being the order of the EAS model. For BSDP, we also take $\mathcal{X}^f[0] := \{298\} \times \{0\}$ for the second-order EAS model (5) and $\mathcal{X}^f[0] := \{298\} \times \{298\} \times \{0\}$ for the third-order EAS model (1), and take $\mathcal{X}^b[N] = \{x \in \mathbb{R}^n | \underline{m}_{\mathrm{NO}_x,\mathrm{tp}} \le x_n \le \overline{m}_{\mathrm{NO}_x,\mathrm{tp}}\}$. We distinguish two cases for BSDP, denoted by BSDP-I and BSDP-II. For BSDP-I, we take $\underline{m}_{\mathrm{NO}_x,\mathrm{tp}} = 0$ kg and $\overline{m}_{\mathrm{NO}_x,\mathrm{tp}} = 28.7 \cdot 10^{-3}$ kg and, for BSDP-II, we take $\underline{m}_{\mathrm{NO}_x,\mathrm{tp}} = \overline{m}_{\mathrm{NO}_x,\mathrm{tp}} = 28.7 \cdot 10^{-3}$ kg. Hence, we force the solution of the latter case to be exactly on the emission legislation limit. The reason for doing so is because of the typical tradeoff between fuel efficiency and engine-out NO_x emissions and exhaust gas temperature (which enhances NO_x conversion in the EAS), which causes the cost optimal solution to lie on the emission legislation target. Choosing $\underline{m}_{\mathrm{NO}_x,\mathrm{tp}} = \overline{m}_{\mathrm{NO}_x,\mathrm{tp}}$ allows us to place the grid points in DP as closely together as possible.

5.2. Optimality of solutions

Our main objective is to minimise the operational costs over the drive cycle. As mentioned earlier, the accuracy of this cost for each DP method depends on the number of grid points p. Therefore, the algorithms of the different DP methods are simulated using the second-order EAS model (5) for different numbers of grid points, i.e., we take $p \in \{10^2, 20^2, 30^2, 50^2, 100^2\}$.

The results of the simulation study are shown in Fig. 5. As BSDP-II with $p = 100^2$ yields to most accurate approximation of the optimal solution, i.e., has the lowest cost, we normalise all results with respect to BSDP-II with $p = 100^2$ for both additional cost and computation time. This means that BSDP-II with $p = 100^2$ has a normalised additional cost of 0. The computation time for the PMP methods is the time needed to find $\lambda[k]$ as in (32), $k \in \{0, 1, \ldots, N-1\}$, that gives the lowest cost satisfying $L(x(N)) \le 0$, while for the DP methods it is the time required to find all optimal control inputs $u^*(x, k)$ as in (13) for all $x \in \mathcal{X}^o[k]$, $k \in \{0, 1, \ldots, N-1\}$. All computations are done on a 2.53 GHz (quadcore) PC, with 4 GB RAM running Matlab 2010b and the computation time for BSDP-II with $p = 100^2$ equals 19 h.

As expected, the computation time and the accuracy increases as p increases. More interestingly, it can be observed that the higher computational complexity of LSDP, ELSDP and BSDP can be compensated by the reduced number of grid points needed to achieve a certain accuracy. The proposed BSDP method has the highest accuracy, which is achieved by including grid points on the boundary of the intersection of the forward and backward reachable sets, while the computational complexity is not significantly higher. Moreover, we can conclude that the real-time implementable solution based on PMP (RT-PMP) achieves a good approximation of the real optimal solution, as the corresponding cost only deviates 0.16% from BSDP-II with $p = 100^2$. The closeness of RT-PMP to the optimal solution demonstrates the need for the proposition of BSDP. Namely, the existing DP methods (i.e., BDP and LSDP) yield a higher cost than the suboptimal RT-PMP solution, when the number of grid points is not sufficiently large.

5.3. Optimal trajectories

We will now discuss the optimal trajectories obtained by the different methods for both the second-order EAS model (5), as well as the original third-order EAS model (1). We use the same

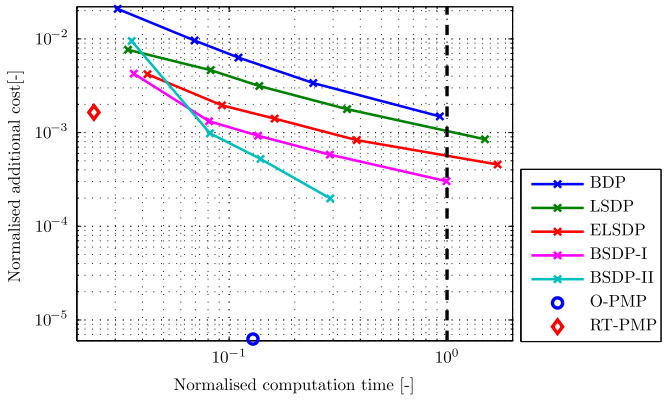


Fig. 5. Computation time versus accuracy of the solution for the second-order EAS model (5).

models for simulating the trajectories as we have used for computing the optimal solution. We will focus for both models on the most accurate DP approach, which is BSDP, the optimal solution obtained using PMP (denoted by O-PMP), and the real-time implementable solution RT-PMP. For the second-order model, we consider BSDP-II with $p = 100^2$, and for the third-order model, we consider BSDP-I, with $p = 10^3$, as the number of grid points cannot be increased due to memory limitations of the PC used in this study. For this third-order model, the upper bound $\overline{m}_{NO_x,tp}$ is also slightly increased in the BSDP computations to ensure that the simulation results do show $m_{NO_x,tp} = 0.8$ g/kWh. Apparently, the interpolation errors that arise in DP with final state constraints are not completely solved in this case.

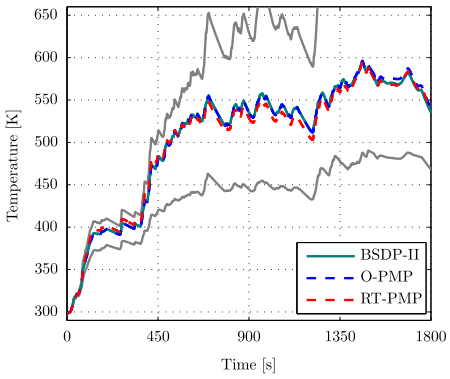
For these three approaches, the EAS temperature T_{EAS} and the cumulative tailpipe and engine-out NO_x emissions $m_{NO_x,tp}$ and $m_{NO_x,eo}$ are shown for the second-order model in Fig. 6, as well as the VTG and EGR mass flows of O-PMP and RT-PMP as a percentage of the respective mass flows of BSDP-II. For the third-order model, the DOC temperature T_{DOC}, SCR temperature T_{SCR}, the cumulative tailpipe NO_x emissions $m_{NO_x,tp}$ and the cumulative engine-out NO_x emissions $m_{NO_x,eo}$ are shown in Fig. 7, as well as the VTG and EGR mass flows of O-PMP and RT-PMP as a percentage of the respective mass flows of BSDP-I. Figs. 6b, c, and 7c, d are normalised over the total work delivered over the cycle. The grey lines in Figs. 6a, b, and 7a–c show the intersection of the reachable sets calculated with BSDP, and the grey lines in Figs. 6c and 7d indicate the maximum and minimum cumulative engine-out NO_x. In Table 2, the additional total fluid cost and the fuel and AdBlue costs, as well as the engine-out emissions are given, where all the costs are normalised with respect to the lowest total fluid cost.

From the simulation results, it can be seen that all solutions aim at increasing the temperatures almost as quickly as possible, see Figs. 6a, 7a and b, while keeping engine-out emissions low (which can be done by using EGR), see Figs. 6c and 7d. From 400 to 600 s, both T_{EAS} and T_{SCR} increases from around 400 K to 500 K, see Figs. 6a and 7a, respectively. This is the range of temperatures where the maximum possible SCR efficiency changes from 60% to 99%, which means that less EGR is required and thus a higher engine-out NO_x can be permitted. This results in lower fluid costs, since (a part of the) expensive fuel associated with EGR is replaced by cheaper AdBlue. In other words, the choice for the inputs is a weighted decision between keeping the SCR efficiency sufficiently high by keeping EAS temperatures sufficiently high (thermal management), while consuming the least amount of fuel in doing so. Note that EGR is used over the complete cycle, as can be concluded from the fact that the achieved engine-out NO_x is lower than the upper bound in Figs. 6c and 7d, which corresponds to using no EGR. This implies that a delicate combination of EGR and SCR results in the lowest possible cost, while satisfying the emission constraints.

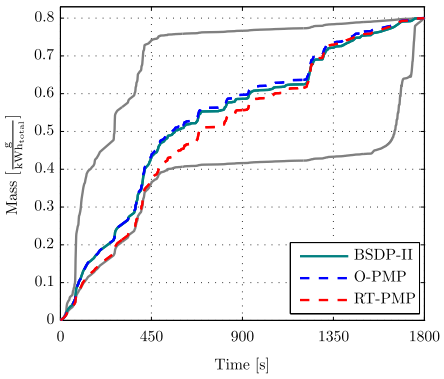
When we compare the different solutions, we can observe in Fig. 6 that BSDP-II and O-PMP result in nearly identical trajectories, while BSDP-I and O-PMP in Fig. 7 are slightly different. This can also be observed from Figs. 6d and e, which show only minor deviations from 100% in VTG and EGR mass flow in the ratio of O-PMP to BSDP-II, respectively, and from Figs. 7e and f, which show somewhat larger deviations from 100% in VTG and EGR mass flow in the ratio of O-PMP to BSDP-I, respectively. The minor differences in the solutions between O-PMP and BSDP-II in Fig. 6 can be explained by the approximations used in determining the solutions. For instance, DP uses linear interpolation for evaluating the cost-to-go, while O-PMP uses a finite difference to approximate the partial derivative $\frac{\partial \eta_{SCR}}{\partial x_1}$ in (32), because an analytical expression for this derivative does not exist. Moreover, the forward–backward sweep method only converges when the number of iterations approaches infinity. The difference between BSDP-I and O-PMP in Fig. 7 is significant after the first 1200 s and can be explained by the limited number of grid points $p = 10^3$ taken to compute the BSDP-I solution, and also the increased grid size because $\underline{m}_{NO_x,tp} = 0$ is used in BSDP-I. The number of grid points used to compute this solution was restricted by the available memory of the PC. This latter fact illustrates the well-known 'curse of dimensionality' of DP. The limited accuracy of the BSDP-I causes the EAS temperature to become higher than necessary, which means that too much energy is spent on heating up the EAS, which results in a slightly higher fluid cost over the cycle, as can be seen in Table 2.

When comparing the results of the second-order EAS model in Fig. 6 and the third-order EAS model in Fig. 7, and the results for engine-out NO_x in Table 2, we can conclude that third-order EAS model predicts a higher overall conversion efficiency, as engine-out NO_x emissions are 3–4% higher, when compared to the second-order model, while the cumulative tailpipe NO_x emissions are the same in all cases. This is due to the fact that the DOC has a relatively small thermal capacity and, hence, the DOC warms up faster than the lumped EAS of the reduced-order EAS model. A fast increase in DOC temperature at the start of the cycle improves the NO_2/NO_x-ratio entering the SCR, which causes it to have a higher efficiency, mainly at lower SCR temperatures. This effect can be seen in the results when comparing the optimal trajectories and the lower bound of the forward reachable sets for cumulative tailpipe NO_x emissions between Figs. 6b and 7c. In the aftertreatment model, the NO_2/NO_x-ratio is captured in the SCR efficiency η_{SCR} in (1), see Willems et al. (2013) for more details. Since the second-order model only has one temperature T_{EAS}, this effect is not taken into account in this reduced-order EAS model. Still, since the model was fitted in Willems et al. (2013) to match experimentally observed temperature profiles, using the second-order model for applications seems reasonable, as NO_x conversion will be underestimated, which causes the emission requirement to be met, when used during experiments.

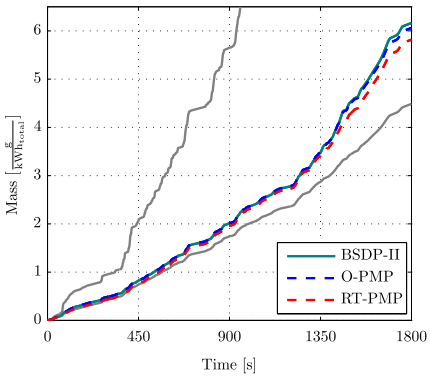
More interesting are the differences between the trajectories of the optimal solutions (BSDP and O-PMP) and the trajectories of RT-PMP. Namely, the study presented in this paper was motivated by the 2.1% fuel consumption and the 1.5% total fluid cost reduction achieved experimentally by IEM in Willems et al. (2013), when compared to baseline strategy, which raised the question how much further IEM can be improved, if a more cost optimal solution strategy would be available. It can be observed from Figs. 6b and 7c that the cumulative tailpipe NO_x trajectory of RT-PMP stays below the trajectories of O-PMP and BSDP during the first 400 s. Moreover, Figs. 6c and 7d show that the trajectories of the



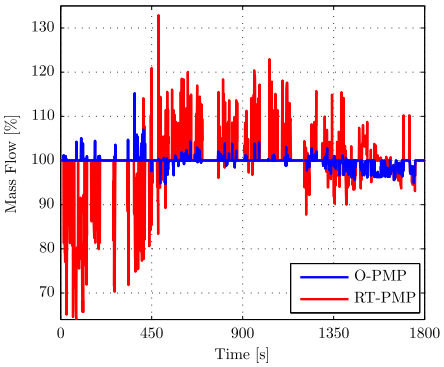
(a) EAS temperature as function of time.



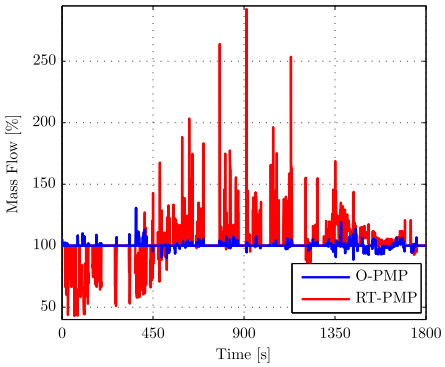
(b) Cumulative NO_x tailpipe emissions as function of time.



(c) Cumulative NO_x engine-out emissions as function of time.



(d) VTG mass flow of O-PMP and RT-PMP as a percentage of VTG mass flow of BSDP-II as function of time.



(e) EGR mass flow of O-PMP and RT-PMP as a percentage of EGR mass flow of BSDP-II as function of time.

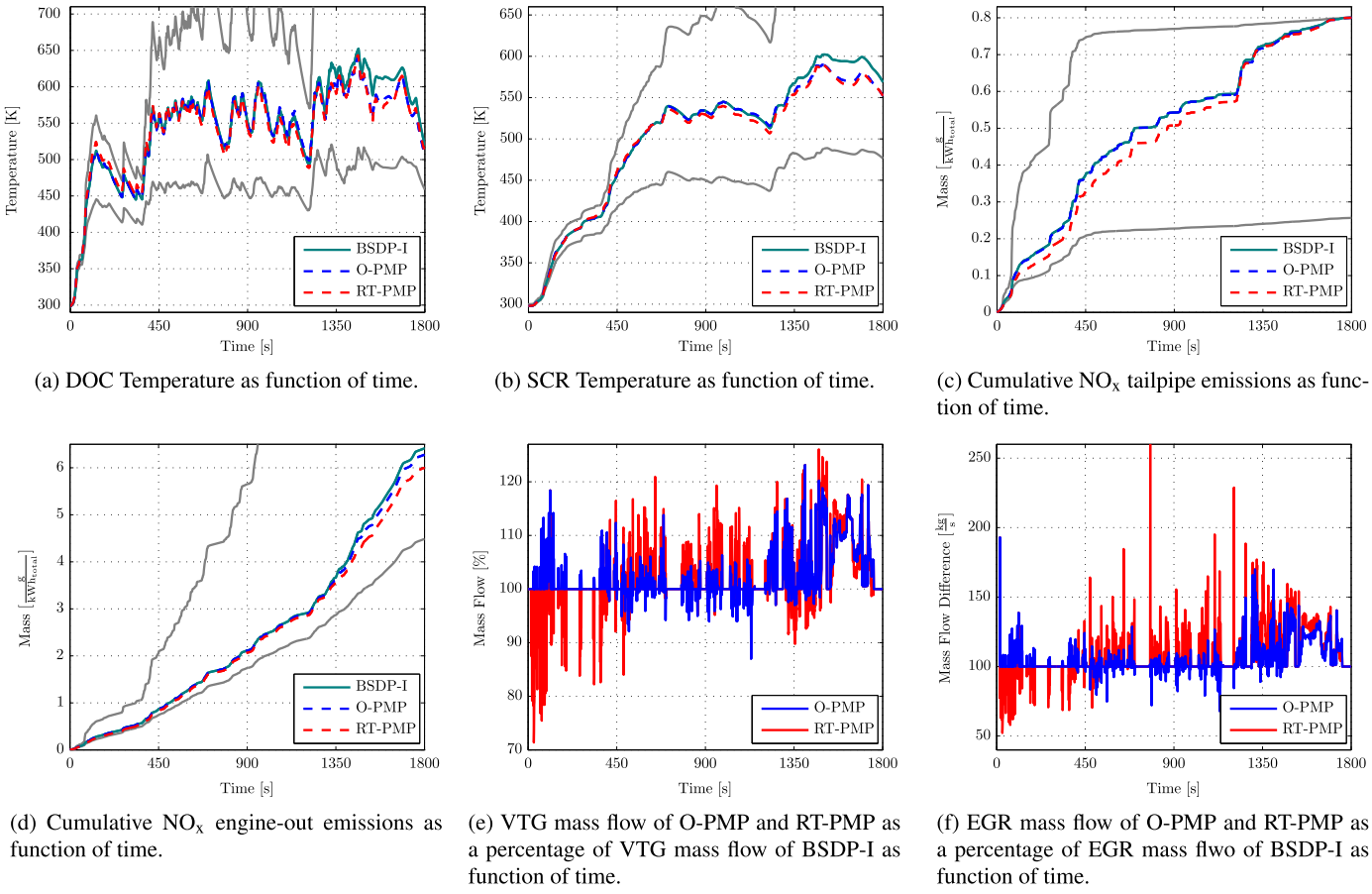
Fig. 6. Results of the Cold-Start WHTC for the second-order EAS model (5).

cumulative engine-out NO_x emissions of RT-PMP even stays close to the lower bound during that period. From Figs. 6d, e, and 7e, f, it can be observed that this is due to using both less VTG and EGR when compared to O-PMP and BSDP. Lower VTG mass flow results in a lower exhaust gas mass flow, which leads to a better NO_x conversion in the SCR (as the exhaust gas resides in the SCR for a larger amount of time), which is useful because of the low EAS temperatures during this time interval. Furthermore, less VTG yields lower engine-out NO_x emissions, as the total exhaust gas mass flow is reduced (while the ratio of NO_x mass flow and total exhaust gas mass flow stays nearly the same). Around 500 s, the difference in cumulative tailpipe emissions is maximal and is around 0.05–0.06 g/kWh. It can be seen that from 500 until 1300 s, T_{EAS} and T_{SCR} for RT-PMP are about 5–8 K lower than T_{EAS} and T_{SCR}, respectively, for BSDP and O-PMP. This implies that the SCR efficiency for RT-PMP is slightly lower than the SCR efficiency for BSDP, meaning that less AdBlue is used than is the case of BSDP and O-PMP, and also implies that EGR is mainly used to stay within emission limits. This can also be concluded from Table 2. To summarise, to be able to satisfy the emission constraints, RT-PMP uses, when compared to BSDP and O-PMP, more of the relatively expensive fuel associated with inefficient engine operation due to not using as much VTG and EGR instead of the cheaper AdBlue. However, the result of this effect is marginal, as can also be seen in Table 2, and RT-PMP approaches the optimal solutions very closely.

6. Conclusions

In this paper, the optimal solution to the Integrated Emission Management (IEM) problem was determined over a cold-start World Harmonised Transient Cycle (WHTC). This optimal solution was determined using both Dynamic Programming (DP) and Pontryagin's Minimum Principle (PMP) and compared to a sub-optimal real-time implementable strategy that was proposed and experimentally validated before. We obtained this solution for a powertrain model that consists of a static engine model and a dynamic engine aftertreatment system (EAS) model. We reduced a third-order EAS model to a second-order model to reduce the computation time in DP. Four different DP methods, two of them extending existing approaches, were discussed, implemented and compared. In addition to DP, the optimal control problem was solved using the application of PMP and using a suboptimal real-time implementable solution.

All proposed methods successfully minimise fuel and AdBlue costs up to a certain numerical accuracy. We found that DP suffers from interpolation errors, but by using Extended Level-Set Dynamic Programming (ELSDP) and Boundary-Surface Dynamic Programming (BSDP) proposed in this paper, we can increase accuracy of the solution and obtain information about the forward and backward reachable sets of the system, at the cost of only a small increase of computational time. The solution obtained through BSDP, where $p = 100^2$ and the terminal constraint is chosen such that the emission legislation is exactly satisfied, delivers the most cost efficient trajectory over the cycle, when the reduced-order dynamic EAS model is considered, while the application of PMP yields the most cost efficient trajectory over the cycle, when the third-order dynamic EAS model is considered, which is caused by the curse of dimensionality of DP. The real-time implementable strategy deviates only 0.16% from the most accurate optimal solution for the case where the reduced-order model is used and 0.08% when the third-order model is considered.



(a) DOC Temperature as function of time.

(b) SCR Temperature as function of time.

(c) Cumulative NO_x tailpipe emissions as function of time.

(d) Cumulative NO_x engine-out emissions as function of time.

(e) VTG mass flow of O-PMP and RT-PMP as a percentage of VTG mass flow of BSDP-I as function of time.

(f) EGR mass flow of O-PMP and RT-PMP as a percentage of EGR mass flwo of BSDP-I as function of time.

Fig. 7. Results of the Cold-Start WHTC for the third-order EAS model (1).

Table 2
Results of the cold-start WHTC for both the second-order and the third-order dynamic EAS model.

EAS Model	Definition	Unit	BSDP	O-PMP	RT-PMP
2nd-order (5)	Addnl total fluid costs	%	0	0.0006	0.16
	Fuel costs	%	98.19	98.22	98.46
	AdBlue costs	%	1.81	1.78	1.70
	Engine-out emissions	g/kWh	6.160	6.060	5.816
3nd-order (1)	Addnl total fluid costs	%	0.23	0	0.08
	Fuel costs	%	98.32	98.14	98.31
	AdBlue costs	%	1.91	1.86	1.77
	Engine-out emissions	g/kWh	6.416	6.276	5.998

The fuel consumption benefits of IEM demonstrated experimentally in Willems et al. (2013) show that IEM is a viable way to contribute to the current challenges of reducing CO_2 emissions. The fact that the real-time implementable strategy deviates only marginally from the optimal solution suggests that further improvements in IEM do not lie in improving the real-time implemental solution strategy, but in improving the control problem formulation. Therefore, future work should focus on improving the models used to compute the control solution and on including additional control inputs (such as injection timing, AdBlue dosing) that can be used for exploiting the synergy between engine and aftertreatment system.

Acknowledgements

The authors would like to thank Frank Kupper of TNO Automotive for fruitful discussions on topic of this paper.

References

Ao, G. Q., Qiang, J. X., Zhong, H., Mao, X. J., Yang, L., & Zhuo B., (2008). Fuel economy and NOx emission potential investigation and trade-off of a hybrid electric vehicle based on dynamic programming. *Proceedings of IMechE, Part D: Journal of Automobile Engineering*, 1851–1864.

Bertsekas, D. P. (2005). *Dynamic programming and optimal control*. Belmont, MA, USA: Athena Scientific.

Cloudt, R., & Willems, F. (2011). Integrated emission management strategy for cost optimal engine-aftertreatment operation. *SAE International Journal of Engines* (pp. 1784–1797), 1784–1797.

de Jager, B., van Keulen, T., & Kessels, J. (2013). *Optimal control of hybrid vehicles*. London, UK: Springer Verlag.

Elbert, P., Ebbesen, S., & Guzella, L. (2013). Implementation of dynamic programming for n-dimensional optimal control problems with final state constraints. *IEEE Transactions on Control Systems Technology* (pp. 924–931), 924–931.

Koot, M., Kessels, J. T. B. A., de Jager, A. G., Heemels, W. P. M. H., Van Den Bosch, P. P. J., & Steinbuch, M. (2005). Energy management strategies for vehicular electric power systems. *IEEE Transactions on Vehicular Technology*, *54*(3), 771–782.

McAsey, M., Mou, L., & Han, W. (2012). Convergence of the forward-backward sweep method in optimal control. *Computational Optimization and Applications* (pp. 207–226), 207–226.

Onori S., & Serrao, L., (2011). On adaptive-ECMS strategies for hybrid electric vehicles. In *International scientific conference on hybrid & electric vehicles*.

Pisu, P., & Rizzoni, G. (2007). A comperative study of supervisory control strategies for hybrid electric vehicles. *IEEE Transactions on Control Systems Technology* (pp. 506–518), 506–518.

Salmasi, F. R. (2007). Control strategies for hybrid electric vehicles: *evolution, classification, comparison and future trends*. *IEEE Transactions on Vehicular Technology* (pp. 2393–2404), 2393–2404.

Sciarretta, A., & Guzzella, L. (2007). Control of hybrid electric vehicles: *optimal energy-management strategies*. *IEEE Control Systems Magazine*, *27*(2), 60–70.

Serrao, L., Sciarretta, A., Grondin, O., Chasse, A., Creff, Y., Di Domenico, D., et al. (2013). Open issues in supervisory control of hybrid electric vehicles: *A unified approach using optimal control methods*. *Oil & Gas Science & Technology*, *68*(1), 23–33.

Sundström, O., Ambühl, D., & Guzzella, L. (2010). On implementation of dynamic programming for optimal control problems with final state constraints. *Oil & Gas Science & Technology* (pp. 91–102), 91–102.

Van Schijndel, J., Donkers, M. C. F., Willems, F. P. T., & Heemels, W. P. M. H., (2014). Dynamic programming for integrated emission management in diesel engines. In *Proceedings of IFAC world congress*, pp. 11860–11865.

Wahlström, J., & Eriksson, L., (2011). Modelling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics. *Proceedings of IMechE, Part D: Journal of Automobile Engineering*, 960–986.

Willems, F., Mentink, P., Kupper, F., & van den Eijnden, E., (2013). Integrated emission management for cost optimal EGR-SCR balancing in diesels. In *Proceedings of IFAC symposium on advances in automotive control*, pp. 711–716.