Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Diss. ETH No. 21593

Optimal Control of Diesel Engines

Modelling,
Numerical Methods,
and Applications

Jonas Asprion
2013

Jonas Asprion

Optimal Control of Diesel Engines

DISS. ETH Nr. 21593

OPTIMAL CONTROL OF DIESEL ENGINES: MODELLING, NUMERICAL METHODS, AND APPLICATIONS

ABHANDLUNG
zur Erlangung des Titels

DOKTOR DER WISSENSCHAFTEN
der
ETH ZÜRICH

(Dr. sc. ETH Zürich)

vorgelegt von

JONAS ASPRION
MSc ETH in Mechanical Engineering
geboren am 17.04.1984
von Escholzmatt, LU

Angenommen auf Antrag von

Prof. Dr. Lino Guzzella, Referent
Prof. Dr. Moritz Diehl, Korreferent

2013

Jonas Asprion
jonas.asprion@alumni.ethz.ch

ETH Zurich
Institute for Dynamic Systems and Control
Sonneggstrasse 3
CH-8092 Zurich
Switzerland

When you truly love something you're doing,
you can't do nothing wrong.
— Ryan Dungey, 2010

To Marieken, Milla and Lilith.

ADDITAMENTUM I.

De Curvis Elaſticis.

I.

JAm pridem ſummi quique Geometræ agnoverunt, Methodi in hoc Libro traditæ non ſolum maximum eſſe uſum in ipſa Analyſi, ſed etiam eam ad reſolutionem Problematum phyſicorum ampliſſimum ſubſidium afferre. Cum enim Mundi univerſi fabrica ſit perfectiſſima, atque a Creatore ſapientiſſimo abſoluta, nihil omnino in mundo contingit, in quo non maximi minimive ratio quæpiam eluceat: quamobrem dubium prorſus eſt nullum, quin omnes Mundi effectus ex cauſis finalibus, ope Methodi maximorum & minimorum æque feliciter determinari queant, atque ex ipſis cauſis efficientibus. Hujus rei vero paſſim tam eximia extant ſpecimina, ut ad veritatis confirmationem pluribus Exemplis omnino non indigeamus; quin potius in hoc erit elaborandum, ut, in quovis Quæſtionum naturalium genere, ea inveſtigetur quantitas, quæ maximum minimumve induat valorem: quod negotium ad Philoſophiam potius quam ad Matheſin pertinere videtur. Cum igitur duplex pateat via effectus Naturæ cognoſcendi; altera per cauſas efficientes, quæ Methodus directa vocari ſolet; altera cauſas finales; Mathematicus utráque pari ſucceſſu utitur. Quando ſcilicet cauſæ efficientes nimis ſunt abſconditæ, finales autem noſtram cognitionem minus effugiunt; per Methodum indirectam Quæſtio ſolet reſolvi: e contrario autem Methodus directa adhibetur, quoties ex cauſis efficientibus effectum definire licet. Inprimis autem opera eſt adhibenda, ut per utramque viam aditus ad Solutionem aperiatur: ſic enim non ſolum altera Solutio per alteram maxime confirmatur, ſed etiam ex utriuſque conſenſu

Leonhard Euler, *Methodus inveniendi lineas curvas maximi minimive proprietate gaudentes sive solutio problematis isoperimetrici latissimo sensu accepti.* Lausannae & Genevae: apud Marcum-Michaelem Bousquet, 1744, p. 245. [http://dx.doi.org/10.3931/e-rara-1490]

Nothing in the world takes place without optimization, and there is no doubt that all aspects of the world that have a rational basis can be explained by optimization methods.⋆

⋆ Free translation by Martin Grötschel in the introduction of *Optimization Stories*, Documenta Mathematica, 2012. The original text is reproduced more closely by the translation provided in J.D. Barrow and F.J. Tipler, *The Anthropic Cosmological Principle*, Clarendon Press, 1986, on page 150: "For since the fabric of the universe is most perfect and the work of a most wise Creator, nothing at all takes place in the universe in which some rule of maximum or minimum does not appear ... there is absolutely no doubt that every affect in the universe can be explained satisfactorily from final causes, by the aid of the method of maxima and minima, as it can be from the effective causes themselves."

Acknowledgements

During the past four years, my life often felt like an optimal control problem. The resource "personal energy" as a state variable needed to be wisely allocated to optimise several contrasting objectives and to sometimes work around seemingly infeasible constraints. Concurrently, the energy had to be regenerated regularly. The problem to be solved thus always was a problem of optimal time management.

In this context, which is nicely captured by Leonhard Euler in his "De Curvis Elasticis", I took the role of the optimisation algorithm that constantly tried to solve the problem "PhD thesis versus the rest of my life". Without the help and the support from numerous persons, this algorithm would have had neither the necessary computational power available nor would it have converged sufficiently fast.

Above all, I thank my wife Marieken. In the presence of "infeasible constraints", she always agreed to slacken her demands to restore feasibility. Furthermore, she always helped me to recharge my personal energy – which also holds for our two children. However, the most important role my wife takes is to keep showing me that some aspects of life do not belong to any optimisation problem but have to be enjoyed just as they come.

Oscar Chinellato, my co-supervisor at FPT Motorenforschung AG, initiated the project "Model-Based Engine Calibration". Besides pro-

viding this fundamental contribution, he kept asking critical questions and often provided ideas on how to tackle certain problems. The discussions with him were always sources of inspiration and motivation, and they helped to keep my "Newton steps" headed in the right direction. Another important person at FPT is Theo Auckenthaler, who initially "hired" me to perform my industrial internship as well as to do the research for my master thesis with the company. It was his persistent and successful pursuit of the model-based approach that paved the way for the realisation of such an innovative project in industry.

The Institute for Dynamic Systems and Control laid the groundwork for this thesis. The lectures held by Prof. Lino Guzzella and Dr. Christopher Onder were some of the best I ever attended, and they inspired me throughout my bachelor and master studies. I am indebted to Prof. Lino Guzzella for providing me with the opportunity to pursue my doctoral studies on a challenging topic. During my doctorate, he focussed on forcing my steps to maintain a reasonable length, even if I could not see any progress at that time. In retrospect, I appreciate this "school of hard knocks" – it forced me to make progress in leaps, not only from a technical point of view, but also personally. I also acknowledge the help I received from Brigitte Rohrbach who copy-edited all my publications. Her knowledge of the English language greatly helped me to improve the quality of my manuscripts.

My co-referee, Prof. Moritz Diehl, had a significant impact on the course of my doctoral studies. In the spring term of 2011 he gave a lecture on "Numerical Optimal Control" at the Institute for Automation in the Department of Electrical Engineering. Until that time I had investigated mainly the heuristic type of optimisation strategies and had only recently discovered the power of the direct methods. His lecture encouraged me to focus my research concerning optimal control on this approach, which turned out to be the perfectly right choice. I am

grateful to him for bringing me into the right “region of attraction”. His fascination for the topic was contagious, and I greatly profited from his profound knowledge.

Last but no least, my colleagues helped me throughout the time at the institute, some by technical or personal discussions, some by an auspicious collaboration. Others managed to act as an example for not taking this part of life overly serious or some step-enforcing measures from the supervisors too personal. In particular, I thank Stephan Zentner for all the joy and the sorrows we shared during our four common years as doctoral students.

J. A.

Abstract

In response to the increasingly stringent emission regulations and a demand for ever lower fuel consumption, diesel engines have become complex systems. The exploitation of any leftover potential during transient operation is crucial. However, even an experienced calibration engineer cannot conceive all the dynamic cross couplings between the many actuators. Therefore, a highly iterative procedure is required to obtain a single engine calibration, which in turn causes a high demand for test-bench time. Physics-based mathematical models and a dynamic optimisation are the tools to alleviate this dilemma. This thesis presents the methods required to implement such an approach. Optimisation-oriented models for the air path and for the in-cylinder processes of diesel engines are derived, and the numerical methods required to solve the corresponding large-scale optimal control problems are presented. The models are shown to be applicable to engines of various sizes and configurations, and a transient experimental validation highlights their accuracy. A self-contained framework for numerical optimal control is implemented. The convergence behaviour and the computational performance of this framework when applied to the problem at hand are analysed. Finally, the resulting optimal control-trajectories over long driving profiles are shown to provide enough information to allow conclusions to be drawn for causal control strate-

gies. Ways of utilising this data are illustrated, which indicate that a fully automated dynamic calibration of the engine control unit is conceivable. An experimental validation of the overall approach demonstrates the applicability of these results. The measurement results show that the dynamic optimisation predicts the reduction of the fuel consumption and the cumulative pollutant emissions with a relative error of around 10% on highly transient driving cycles.

Zusammenfassung

Um die stetig strenger werdenden Emissionsvorschriften zu erfüllen und gleichzeitig dem Wunsch nach noch geringerem Treibstoffverbrauch nachzukommen, muss der transiente Betrieb von Dieselmotoren optimiert werden. Um das verbleibende Potential auszuschöpfen, stellt ein aktueller Dieselmotor viele, sich jedoch gegenseitig beeinflussende Aktuatoren zur Verfügung. Selbst ein erfahrener Applikationsingenieur benötigt viele Iterationen, um eine akzeptable Bedatung aller Kennfelder und Reglerstrukturen für dieses komplexe System zu erlangen. Da komplette dynamische Fahrzyklen betrachtet werden müssen, resultiert ein immenser Bedarf an Prüfstandszeit. Das Ziel dieser Arbeit ist die Erstellung eines Werkzeugs, das den Ingenieur insbesondere für die Applikation des transienten Betriebes unterstützt und somit die kostenintensive Prüfstandszeit reduziert. Auf physikalischen Grundgesetzen bauende, mathematische Modelle für alle Prozesse im Motor sind dazu genauso nötig wie eine numerische Methodik, um schnell und verlässlich dynamische Optimierungsprobleme zu lösen. Die Modelle werden für sich in Grösse und Aufbau unterscheidende Motoren entwickelt und für den transienten Motorbetrieb validiert. Als zweiter Baustein wird ein Framework für die numerische Berechnung von optimalen Steuergrössenverläufen entwickelt und das Optimierungsproblem für Dieselmotoren entsprechend formuliert. Das Vorgehen

ermöglicht die dynamische Optimierung der Steuergrössen über lange Fahrprofile, wodurch genügend Information generiert wird, um daraus kausale Regelstrategien abzuleiten. Es wird aufgezeigt, wie die Resultate der Optimierung verwendet werden können, um automatisch eine für transienten Betrieb optimierte Applikation des Motorsteuergerätes zu erhalten. Die experimentell festgestellte Verlässlichkeit der Voraussagen der Optimierung verleiht diesen Resultaten den nötigen Bezug zum realen Motor.

Contents

List of Figures

List of Tables

1 Introduction

Optimal control of diesel engines becomes increasingly important. More stringent emission regulations require the exploitation of the remaining potential of reducing the emissions, not only in stationary operation but especially during transients. Simultaneously, the fuel consumption has to be minimised for economical and environmental reasons. Current and near-future pollutant limits [92, 157] guide the attention towards transient operation [127, 94].

The classical approach to parameterise an engine-control unit (ECU) is to first derive stationary lookup maps. Subsequently, transient corrections are added to these static maps for the control inputs and the feedback variables, and heuristic feedforward parts are included. This approach, which mainly relies on engineering experience, leads to a highly iterative procedure when transient driving cycles are considered. With the increasing complexity of the engine systems, it becomes difficult for the calibration engineer to conceive all the cross-couplings between the many actuators. A high demand for testbench time results. Furthermore, each new calibration of an engine requires this manual procedure to be executed again. Especially for heavy-duty en-

gines, which are usually applied to a variety of tasks, a fast and partially automated calibration process is desirable.

Model-based approaches and mathematical optimisation tools provide the means for such an automation. For example, during the calibration of the ECU, the static lookup maps need to be optimised. This optimisation can be performed directly on the engine [52], or mathematical models may be used in place of the physical engine [93, 24, 91]. An overview on modelling approaches and the corresponding data acquisition can be found in [44]. The information about the driving cycle to be considered may be included by weighting the operating range of the engine. This weighting matrix represents the time during which the engine is operated at a certain engine speed and load. Alternatively, a few relevant operating points can be identified, which allows the optimisation to be restricted to those points [10, Ch. 7]. Identifying the parameters of a feedback controller can also be cast as an optimisation problem. Again, the solution can be obtained directly on the engine [57] or by a model-based approach [179]. Adaptive methods try to automatically improve the engine calibration during normal operation [131].

The effects of the air-path dynamics may be incorporated either online by model-predictive control [73, 72, 62], or offline by an optimisation of the control trajectories over a representative driving profile. The solution of such optimal control problems (OCPs), which in the past were also obtained directly on the engine [66], provide implications for well-suited control structures [174, 186, 70, 33]. Namely, the optimal control-input trajectories with the corresponding data on fuel consumption and pollutant emissions can be used as a benchmark to disclose scenarios in which the performance of actual control systems is suboptimal. An analysis of these cases provides insights into the way the control structure, the feedforward control, and the reference-trajectory

generation can be improved. Another way of utilising the data obtained from the solution of OCPs is to train function approximators such as artificial neural networks [146, 145].

In the literature referenced, the authors have stated that due to the complexity of the system, OCPs for diesel engines can only be solved over short time horizons. However, in order to derive implications on a suitable control structure or to train function approximators, the optimal solution over long horizons is necessary. In this thesis, the means to achieve this goal are presented.

During an optimisation, a vast number of model evaluations are performed. Either the model function itself is evaluated, e.g. during static calibration or when simultaneous methods for optimal control [114] are employed, or a forward simulation of the model is used, e.g. for parametric studies or in the context of shooting methods for optimal control [37, Sec. 3.2–3.4]. Thus, the model has to be simple in order to allow a fast execution. In both cases, partial derivatives need to be calculated as well. If only standard mathematical operations are used, automatic differentiation is applicable, enabling a fast and accurate evaluation of partial derivatives [158, 88].

Furthermore, models apt for optimisation have to be smooth and quantitatively accurate, they must capture all relevant qualitative trends and allow for plausible extrapolation. These properties make the model itself or its simulation an “accurate and consistent function generator” [37, Sec. 3.8]. Furthermore, such models improve the convergence properties of the optimisation by eliminating the need for factitious constraints and by providing the problem with an inherent stability. As a fundamental requirement, the model outputs have to be predicted using the control signals and the ambient conditions only. Ideally, the model requires a small set of measurements for its identification in order not to cancel the reduction in test-bench time gained by the

application of model-based approaches. If it is capable of predicting the effects of influences that cannot be measured easily, its value for parametric studies is further enhanced.

Optimisation-oriented models are thus classified between control-oriented models and phenomenological models [155, 18, Sec.1.1]. The former are models that can capture the trends relevant for control while being as simple as to be implementable even on an ECU. Accuracy is of minor importance due to the presence of feedback control, and they only have to be valid in a region around the setpoints of the controller. In contrast, phenomenological models are mainly used to perform predictive parametric studies or to analyse specific effects. The execution speed is not critical in that context, such that one evaluation typically takes from a few seconds up to several hours [148]. Due to their complexity, such models often suffer from error propagation.

1.1 The CTI project

This thesis is partially funded by the Swiss Innovation Promotion Agency CTI under grant no. 10808.1 PFIW-IW. The motivation of the project stems from the need of the industry for a tool to support the calibration engineers. Accordingly, the project was initiated by the industrial partner, the FPT Motorenforschung AG in Arbon, Switzerland. As described above, tools for the stationary calibration of the ECU are readily available. However, due to the increasing complexity of the engine systems, the tuning of the dynamic behaviour of the engine is a task which can hardly be tackled by an engineer relying only on his experience. The multitude of control inputs and the cross-influences between them require a tedious and highly iterative procedure to obtain a good calibration of all dynamic features of the control system. Throughout this process, the engine needs to be operated on the engine test bench, which becomes an increasingly critical cost factor in the calibration process.

In order to cut the test-bench time and to enable a faster dynamic engine calibration, a model-based approach is desirable. A mathematical model replaces the physical engine. The model runs faster than real time, can be executed in parallel on several processing units, cannot fail or be damaged, does not consume any fuel, and does not require any staff that operates and maintains the physical test bench, the measurement devices and the engine. The goals of this project represent the first steps towards the implementation of a fully automated, model-based approach. They comprise the development of optimisation-oriented models for diesel engines and the assessment of methodologies to address the large-scale OCPs to be solved. The most promising approach should be developed to a point that enables the evaluation of the overall methodology. Furthermore, first tests on the utilisation of the data obtained from the solution of the OCPs should be performed.

1.2 Informal outline

At the outset of the project, a simple model for the engine was used. Standard model components for the air path were combined with empirical models for the torque generation and the pollutant emissions. The broad field of optimisation was analysed and various approaches were tested, e.g. heuristic methods such as iterative dynamic programming [130] or approximate dynamic programming [153, 181]. Classical dynamic programming, as well as indirect approaches such as the Pontryagin minimum principle cannot be applied due to the dimensionality and the complexity of the problem. Finally, direct methods are found to be the most suitable approach to solve the problem at hand. These methods first discretise the continous problem and then solve the resulting large, but finite optimisation problem by applying Newton-type solvers. An overview of the approaches for solving optimal control problems, the choice of a specific method, and all aspects of the method developed and scrutinised during this project are provided in Ch. 3. A complete framework for numerical optimal control was developed, which is described in Appendix D.

Once the method used for the solution of the OCP was selected, an engine model suitable for this approach had to be developed. As mentioned above, the model has to feature a fast and reliable execution, smoothness, and a plausible behaviour for interpolation and extrapolation. The other main workload during the project thus was to develop a control-oriented model for full diesel-engine systems. The model was developed and tested on various engines, for which the measurement data was obtained on test benches provided by the industrial partner. The data processing and the calculation of relevant quantities from the raw data are presented in Appendices B and C. The models for the air path, the cylinder processes, the torque generation, and the pollutant emissions are described in Ch. 2.

Finally, specific properties of the OCP of diesel engines needed to be considered. The problem formulation, its initialisation, as well as several specific aspects of this problem are presented in Ch. 4. The results are described in Ch. 5. The computational performance of the optimisation algorithms is analysed, and the properties of various variants of the chosen discretisation method are investigated. Furthermore, several case studies concerning the engineering aspects of the problem are presented. These applications illustrate ways of utilising the optimal control methodology to assist the calibration engineer and to even partially automate the calibration process.

1.3 Scientific contributions

The scientific contributions of this thesis are summarised in the following list.

1. *Modelling of diesel engines*
   (a) Development of a physics-based model for the NO_x emissions that is suitable for optimal control [18].
   (b) Development of physics-based models for all in-cylinder processes such as the compression and the ignition delay, as well as for the torque generation [19].
   (c) Implementation of a completely smooth and algorithmically differentiable engine model and its validation during transient operation [19].
2. *Numerical optimal control*
   (a) Development of a framework for the direct numerical solution of optimal control problems using Radau collocation. The equivalence of the integral and the differential formulations of this scheme is exploited to construct a consistently discretised problem.
   (b) Application of this framework to a problem not related to this project [11].
   (c) Development of algorithms for the iterative solution of time-subdivided large-scale optimal control problems with integral constraints.
   (d) Exploitation of the sparsity structure of the nonlinear program resulting from the application of direct collocation to optimal control problems to perform a partitioned quasi-Newton update of the Hessian of the Lagrangian.

3. *Optimal control of diesel engines*

   (a) Calculation of optimal control and state trajectories on long driving profiles by an efficient implementation of the direct collocation approach [20].

   (b) Utilization of this optimal solution to perform a completely automatic calibration of a causal controller for the combustion and the air path, including feedback control [20].

   (c) Application of the time-subdivision approach to the diesel-engine problem.

   (d) Consistent formulation of the optimal control problem for diesel engines, including drag phases [22].

   (e) Comparison of static and dynamic control-input optimisation strategies, and derivation of implications for the control structure from these solutions [21].

   (f) Development of an iterative procedure for the solution of the optimal control problem directly on the engine. Specific time-varying models for the combustion processes and a dynamic refinement of the air-path model are derived [17].

1.4 Engines used in this work

The main data of all diesel engines used in this work is listed in Table 1.1. All engines are equipped with a single-stage, variable-geometry turbine (VGT) turbocharger and a common-rail injection system. The type of the engine and the test bench, as well as an outline of the tasks for which each engine was used follows.

A Heavy-duty engine, industrial test bench. The development of the NO_x model and of the air-path model were performed using mainly data from this engine. Furthermore, since the model for this engine was the first one available, it was used for most of the numerical tests and the parametric studies concerning the optimal-control framework.

The experimental validation of the optimal control trajectories was performed on the same engine type. However, that engine was equipped with a slightly different injection system and a different turbocharger. A different ECU calibration was used as well.

B Light-duty engine, industrial test bench. The version without EGR was used along with engine A during the development of the NO_x model and served to assess the generality of the air-path model components. The version with EGR was used to extend the models for the air path, the cylinder processes, and the NO_x emissions to the case with EGR.

C Heavy-duty engine, research test bench. Only a finite set of measurement data was available from a prior project at the institute. However, since the measurements required also contain EGR variations, that data was crucial for the initial development of the NO_x model.

D Passenger-car and light-duty engine, research test bench. This engine was set up on a test bench at the institute. The continuous availability of the engine and the flexibility of the test-bench setup enabled the development and a first implementation of the iterative procedure to solve the dynamic optimisation problem.

E Heavy-duty engine, industrial test bench. The measurement data used to analyse the thermal processes and to develop the corresponding models were obtained on this engine.

Table 1.1: Main data of the engines used. "I" denotes an in-line cylinder configuration, as opposed to a "V" engine. Only high-pressure (HP) EGR systems were available.

Engine:		*A*	*B*	*C*	*D*	*E*
displ. volume V_d	[l]	8.7	3.0	6.4	3.0	12.9
cylinders n_{cyl}	[-]	6 (I)	4 (I)	6 (I)	6 (V)	6 (I)
V_d / n_{cyl}	[l]	1.45	0.75	1.06	0.50	2.15
bore	[mm]	117	96	102	83	135
stroke	[mm]	135	104	130	92	150
compression ratio	[-]	15.8	17.6	18.1	15.5	17
EGR		-	(HP)[a]	HP	HP	-
rated power	[kW]	300	130	210	165	413
→ at	[rpm]	1'650	2'900	2'200	3'800	1'575
max. torque	[Nm]	1'720	420	1'120	540	2'500
→ from	[rpm]	1'200	1'400	1'200	1'600	1'000

[a] For engine B, versions without and with EGR were available.

2 Modelling of diesel engines

Modelling a turbocharged diesel engine is a challenging task. The engine represents a complex, highly nonlinear dynamic system with at least one dynamic feedback loop, namely the turbocharger. The enthalpy in the exhaust gas is extracted by the turbine to speed up the turbocharger. This mechanical energy is transferred to the intake air by the compressor to drive a mass flow against a pressure rise. This mass flow increases the pressure in the intake manifold. Since the temperature is held approximately constant by the intercooler, the density of the gas in the intake manifold increases, leading to a higher mass flow into the cylinders. In turn, this higher flow provides more enthalpy to the turbine after being heated by the combustion.

In an engine equipped with an exhaus-gas recirculation system (EGR), an additional feedback loop is introduced. This loop is coupled with the turbocharger loop. The EGR system enables a mass flow from the exhaust to the intake, which is usually extracted before the turbine (high-pressure EGR). The enthalpy of this EGR mass flow is not available to the turbine to drive the turbocharger. On the other hand, it increases

the pressure in the intake manifold and thus also the mass flow into the cylinders.

Due to this coupling, the effects of the two actuators associated with the turbocharger and the EGR depend on each other. The EGR valve controls the EGR mass flow. A controlled turbocharger either has a wastegate, which bypasses a fraction of the exhaust-gas flow past the turbine, or a variable-geometry turbine (VGT). The latter allows the opening area of the turbine to be modified in order to change the restriction of the flow. The physical engine layout shown in Fig. 2.1, or the more abstract ones in Fig. 2.5 may help to visualise the dynamic cross couplings between the air-path actuators.

The physical processes in the air path can be captured by simple thermodynamic and fluiddynamic relations. Conversely, the combustion and all related phenomena in the cylinders are complex and hard to describe by physical first principles. However, since the combustion is a fast process and the combustion events occur at a high frequency, a common assumption is that this part of the system can be considered an instantaneous process. Especially when compared to the slow turbocharger dynamics, which have a characteristic time of a few seconds, this assumption is reasonable. Figure 2.2 illustrates the subdivision of the overall model of the engine into a dynamic and a static part. In order to limit the propagation and amplification of errors, the two parts should be matched by a sensitivity analysis.

When deriving an engine model, its intended use has to be kept in mind. For example, a model suitable for conceptional studies concerning the combustion chamber geometry has to live up to different requirements than a linearised, control-oriented model that runs online on the engine control unit. Models apt for optimal control have to combine several aspects of both extremes. For the results of the optimisation to be of any relevance, the model on which the optimisation

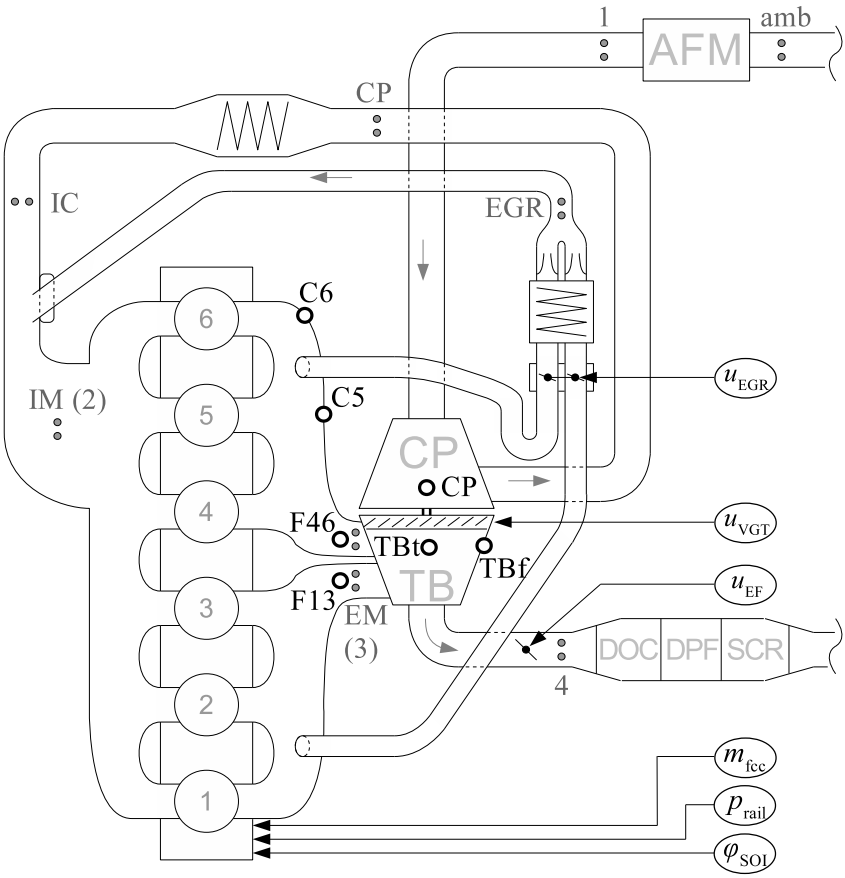

Figure 2.1: General layout of a diesel engine. An in-line 6-cylinder engine block with a two-fluted exhaust system is shown. The grey dots indicate the locations of temperature and pressure sensors, whereas the circles represent the placement of surface-temperature sensors (C: cylinder, F: flute, t/f: top/front). All control inputs considered in this work are shown (encircled).

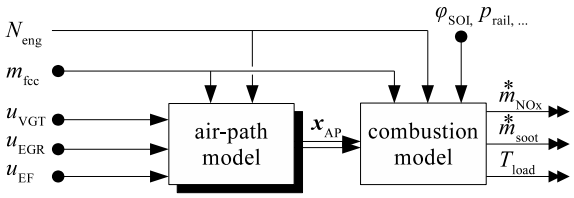

Figure 2.2: Subdivision of the engine model into a dynamic and a static part. Several more control inputs to the combustion submodel could be included. In some engines, additional air-path actuators are present such as a swirl valve, additional multi-stage or parallel turbochargers, a low-pressure EGR path, a variable-geometry compressor, or a throttle in the intake. The air-path actuators affect the combustion and thus the torque production and the pollutant emissions indirectly by the state variables of the air path $\boldsymbol{x}_{AP}$.

relies has to be accurate and must capture all influences of each control input. The vast amount of model evaluations during the optimisation require the model to be fast, which is primarily achieved by keeping the structure simple. Finally, the model has to be smooth since an efficient optimisation relies on (first and even second) derivatives. Especially the combustion processes are hard to describe by models that combine all those requirements.

The models for the air path and the torque generation are based on [92]. All other related work that has provided specific ideas or modelling components is referenced at the appropriate places in the text. The remainder of this chapter is structured as follows. After a short description of the measurement data used, the air-path model is presented. Its structure is derived from the physical engine layout shown in Fig. 2.1, and model versions of various complexity levels are introduced. Subsequently, the models for the torque generation and for the pollutant emissions are developed. The latter focus on the emissions of nitrogen oxides (NO_x), which is one main contribution of this thesis. The val-

idation of each model component is presented within the respective sections. Once all individual parts are introduced, an assessment of the overall model is presented to conclude the chapter.

2.1 Measurement data

Since engines A, B and E were set up on fully automated test benches, finely resolved maps were recorded and specific variations were performed for these engines. For engine C, only the data from a sparse variation scheme collected for a previous project was available. However, this data set was crucial since engine C initially was the only one with an EGR system.[1] Exhaust-gas recirculation explicitly changes the intake-gas composition and thus enables an unambiguous identification of the activation temperature in the NO_x-formation model. The research test-bench of engine D does not provide an automatisation of stationary measurements. However, the interface is more flexible and thus, the specific transient tests for the iterative model refinement were performed on this engine, see Sec. 5.2.1. Only the stationary measurements necessary for a coarse identification of the air-path model for this engine were performed.

Two sets of stationary measurements were conducted on engines A and B. A base map was measured on a dense grid (with 150 to 260 points) covering the full operating range of the engine. This data is termed “engine map” in the remainder of this work. As control inputs, the values provided by engine calibrations close to series-production specifications were used. The coverage of the full operating range provides information for conditions with an extremely low mass flow and almost no fuel injection. This data is crucial for obtaining a model that is able to capture the engine behaviour near idling conditions and during drag phases.

Variations from the setpoints defined by the engine calibration were measured in 16 operating points spread over the full operating range, as shown in the left-hand plots of Fig. 2.3. In these “variation points”, a

[1] The configuration of engine B with EGR became available only towards the end of the project.

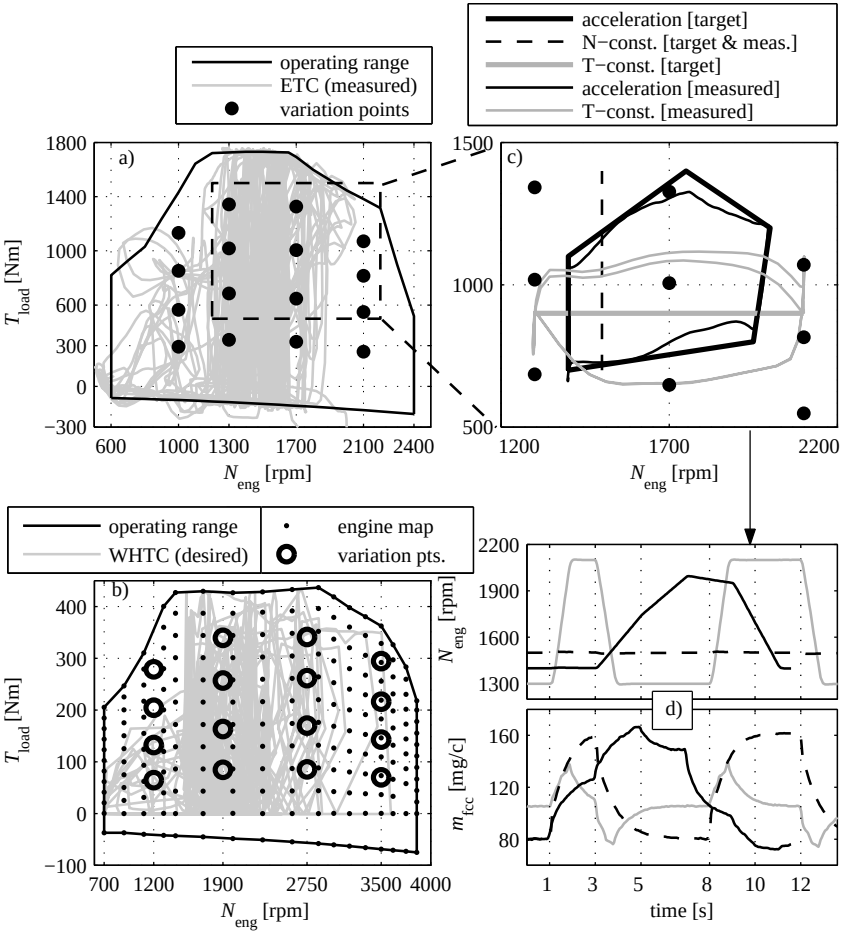

Figure 2.3: Measurements used for the model development and validation. a) Speed-torque map of engine A, with variation points and trajectory of a measured European Transient Cycle (ETC). Engine map not shown to increase clarity of the plot. b) Speed-torque map of engine B, with variation points, engine map, and desired profile of the World-Harmonized Transient Cycle (WHTC). c) Targets and effectively achieved trajectories of the transient test-cycles used for the validation of the NO_x model. d) Measured engine speed and fuel injection for the test cycles shown in plot c).

detailed sweep through each individual control input, namely the VGT position, the injection timing and the rail pressure, as well as two- and three-fold cross-variations were performed. The variation space was chosen as large as possible, and was constrained only by mechanical and thermal limits such as the maximum rail pressure, cylinder peak-pressure, turbocharger speed and exhaust-gas temperature as well as the requirement of a stable engine operation. For engine B with EGR, also the EGR rate was varied by a sweep through the full range of the EGR valve from closed to fully open, followed by a step-wise closing of the exhaust flap. Moreover, variations of the injection timing and the rail pressure were performed while applying the setpoint values for the EGR valve and the exhaust flap. Finally, isolated sweeps of the exhaust flap were recorded.

The variation of the injection timing was large enough to enter the so-called "NO_x-bump" region [140] in both the late and the early direction. This region is characterised by exceptionally long ignition delays, causing the combustion to be almost fully premixed, which contradicts a basic assumption of the NO_x model, see Sec. 2.5.3. Since such large premixed-combustion fractions result in a fast and thus hot diffusion combustion, the NO_x emissions increase again, as opposed to the expected trend. Moreover, due to the inapt combustion timing, a simultaneous decrease in efficiency and thus an increase in fuel consumption results. Therefore, it is not advisable to enter these regions, and the corresponding measurements were excluded from the data for the identification of the NO_x model and the combustion efficiency (but not for the identification of the air path and the cylinder models). The resulting upper and lower limits for the SOI can be closely approximated by an engine-specific linear function of the engine speed. These limits have to be included in the formulation of the optimal control problem to ensure meaningful results.

Transient measurements Transient data was obtained by running various homologation cycles using the current control structure and engine calibration. Moreover, specific measurements for the development and the identification of the thermal model were performed on engine E. These measurements are described in Sec. 2.2.3. Finally, to enable a thorough transient validation of the NO_x model, specific transient measurements were performed on engine A. Three different test cycles were measured, each subject to a set of input variations. Two synthetic cycles, namely holding the engine speed or the load torque constant, were complemented by a cycle inspired by a recurring pattern in the European Transient Cycle (ETC) [157]. It approximates one up-shifting interval during acceleration. The desired and the actually achieved speed-load trajectories of the three cycles are shown in the top right-hand plot of Fig. 2.3. The trajectories are chosen intentionally to lie between the variation operating-points and to cover ETC-relevant regions of the operating range. The deviations from the desired profiles are caused by the torque controller, which is not able to adjust the injection quantity sufficiently fast during forced changes of the engine speed. The trajectories of the engine speed and the injection quantity actually measured are depicted in the bottom right-hand plot of Fig. 2.3.

Due to the turbocharger lag, the boost level inherently deviates from its stationary value during transients. Therefore, only the injection timing and the rail pressure were explicitly varied. The pressures in the intake and exhaust manifolds deviate by up to 600 mbar from their stationary value, often in opposite directions. The resulting even larger deviations in the pressure difference over the engine cause values for the residual-gas fraction and the cylinder charge which cannot be generated by stationary variations. Therefore, a substantial extrapolation has to be accomplished by the model. After compensation of the sensor dynamics as described in Sec. 2.1, the temperatures of the intake air

and the exhaust gas differ from their stationary values by up to 8K and 150K, respectively.

The variation grid for the injection timing and the rail pressure consisted of a full 3×3 matrix, with the center point being the setpoint, and variations of $\{-4,0,+4\}$°CA × $\{-250,0,+250\}$ bar. These were the maximum allowable values still guaranteeing a safe engine operation during transients. The four intermediate points given by the variations $\{-2,+2\}$°CA×$\{-125,+125\}$ bar were added to the variation schedule. For a fast recording of the NO_x emissions, the sampling head of a Cambustion fNOx400 analyser with an NO_2 converter was installed directly after the turbine.

Data processing The measured cylinder-pressure signal is processed as described in Appendix C. The signal provided by the fNOx400 analyser is distorted by offsets due to sensor drift and the manual calibration procedure between any two measurements.[2] A stationary accurate, but slow measurement of the NO_x concentration was available, and this signal was used to match the fast fNOx400 signal. Furthermore, a moving-average filter with a window of 80 ms had to be applied to the raw fNOx400 data to suppress noise and to eliminate cylinder-to-cylinder variations.

Several signals had to be compensated for dynamic effects observed during transient engine operation. The air flow into the cylinders is not equal to the flow measured by the air-flow meter (AFM) that is located before the air filter. Since the pressure and the temperature between the AFM and the compressor are almost constant, the flow into the compressor is equal to the measured flow. However, the pressure in the

[2] The sensor drift caused by probe fouling necessitates a frequent re-calibration of the device when installed on a diesel engine. Thereby, the zero level and the span of the fNOx400 analyser have to be calibrated manually by switching between reference gases and adjusting the corresponding output using the rack display.

intake manifold varies significantly. When it rises, the mass stored in the manifold increases, resulting in the actual flow into the cylinders being smaller than the flow through the compressor. The temperature may be approximated as being constant due to the intercooler, yielding the compensation

$$\overset{*}{m}_{\mathrm{cyl}} \approx \overset{*}{m}_{\mathrm{AFM}} - \frac{d}{dt} p_{\mathrm{IM}}(t) \cdot \frac{V_{\mathrm{IM}}}{R \cdot \vartheta_{\mathrm{IM}}}. \tag{2.1}$$

The temperature in the intake manifold is measured by a slow 3mm thermocouple. The physical quantity may be reconstructed from this measurement signal. Representing the noisy signal by splines enables an analytical calculation of its derivative. This smooth derivative is used to invert a first-order element with a time constant of 3.5 s, which is identified by comparison to measurement data obtained with a fast 0.5 mm thermocouple. The intake temperature rises due to an increasing compression of the air and thus exhibits a shape similar to that of the slowly changing turbocharger speed. Therefore, the smooth shape resulting from the compensation procedure described above can be assumed to accurately represent the transient evolution of the intake temperature.

The situation in the exhaust manifold is more difficult. High temperatures and variable heat losses to the manifold walls additionally distort the measurement results. Moreover, the reconstructed signal cannot be assumed to be smooth since, for example, a sudden change of the injection quantity leads to an instantaneous change of the temperature of the exhaust gas. Therefore, this temperature was calculated using the corresponding air-path submodel, which is identified using stationary measurements only.

2.2 Mean-value model of the air path

This section describes the model of the air path of diesel engines. First, an engine layout without EGR is considered, which is subsequently extended by an EGR system. Figure 2.4 shows the cause-and-effect diagram of a general air-path model.

A first simplification is the omission of the volume between the compressor and the intercooler. The pressure ratio over the intercooler hardly ever exceeds a value of 1.05. Therefore, this pressure drop is attributed to the compressor, over which much larger pressure ratios occur. To achieve a consistent model, the pressure ratio over the compressor is calculated using the intake-manifold pressure instead of the pressure before the intercooler during the model identification.

The model can be simplified further by omitting the restrictions imposed by the air-flow meter (AFM) and the aftertreatment system (ATS). Thus, the pressure and temperature before the compressor and after the turbine correspond to the ambient conditions. During the model identification, this structure is taken into account, i.e. the compressor and the turbine models are "misused" to also account for the pressure drops over the respective restrictions. This simplification removes two state variables and two restrictions from the model. However, this advantage is available only in exchange for a less general model. Namely, the model is not able to predict the influence of changing ambient conditions anymore. The effect of this simplification is analysed in greater detail below.

The two model versions just explained are shown in Fig. 2.5. As a third model version, the one representing an engine with EGR is considered. The diagrams follow the gas flows through the engine rather than representing the physical causality as it is the case in Fig. 2.4. Each model version evolves from the previous one by extending it by

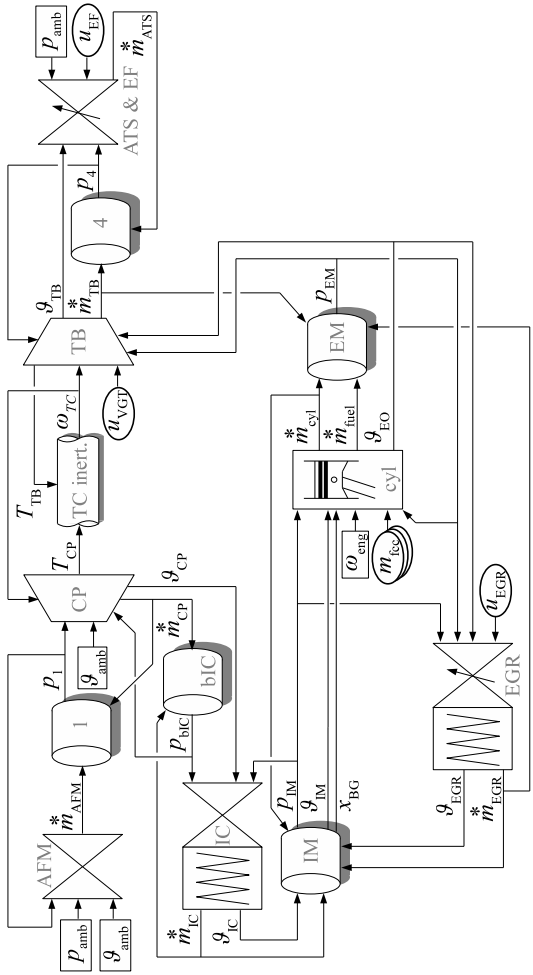

Figure 2.4: Cause-and-effect diagram of a general air-path model. Multiple control inputs into the cylinder block are indicated because also the combustion control (e.g. start of injection) has a slight influence on the exhaust-gas temperature.

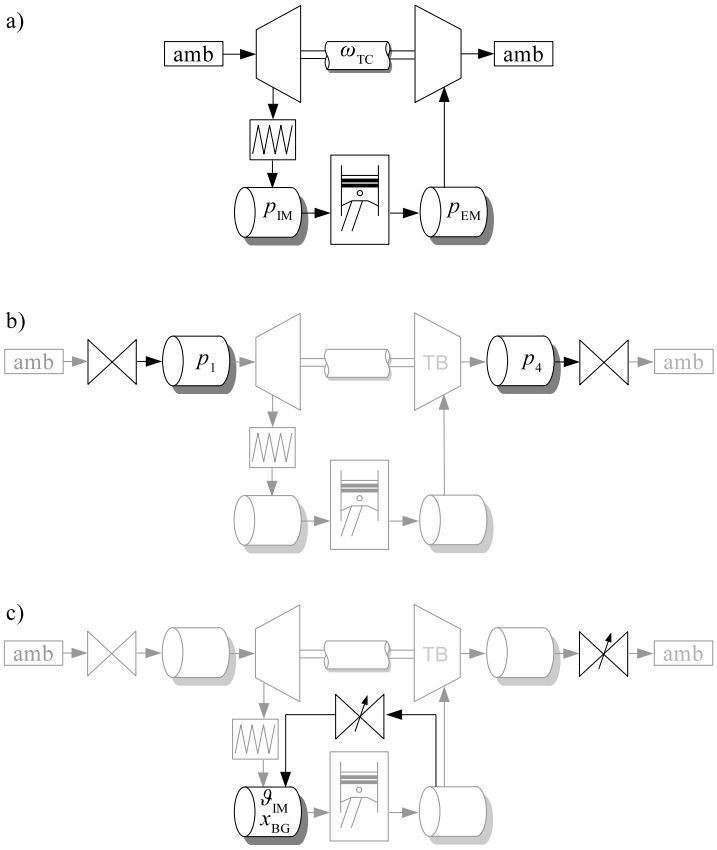

Figure 2.5: The three model versions used in this work. The simplest one represented by a) has 3 state variables, namely the turbocharger speed and the pressures in the intake and the exhaust manifolds. By explicitly accounting for the influence of the flow restrictions in the intake and the exhaust systems, the structure b) results. The pressures in the two additional receivers introduce two additional state variables to the model. Finally, c) represents the air-path model for an engine with EGR. In the intake manifold, balance equations for the temperature and the burnt-gas fraction have to be added, introducing another two state variables to the model.

additional components and state variables. To clarify these changes, only the new or modified parts are highlighted in Fig. 2.5.

In the remainder of this section, first the submodels for all common parts are derived. Subsequently, the specific components of an engine with EGR are modelled, and a model for the thermal state of the engine is developed. It will be shown that the latter has a substantial impact on the accuracy of the air-path model. Finally, a validation of the overall air-path models during transient operation of the engine concludes this section.

2.2.1 Air-path components

This section describes the model components that are common for the engine with and without EGR. After the description of each submodel, the procedure for its identification is outlined.

Compressor mass-flow The model for the mass flow driven by the compressor is based on [124]. Surge is not considered since after the solution of an optimal control problem, the engine has to be operated strictly out of the surge region of the compressor. During optimisation, a stable behaviour has to be guaranteed. The speed lines need to have a positive slope towards lower mass flows, up to zero mass flow. This characteristic leads to small mass flows for overestimated pressure ratios, which in turn reduces the pressure ratio again.

In contrast to the original literature, the ellipse equation is solved for the mass flow instead of the pressure ratio. The latter is appropriate only when a surge model is included. The equations of the original

model are listed below, where the index s denotes the surge limit.

$$\Pi_\mathrm{s} = 1 + k_{\pi,1} \cdot N_\mathrm{TC}^{k_{\pi,e}}, \tag{2.2a}$$

$$\overset{*}{m}_\mathrm{s} = 0 + k_{\mathrm{m},1} \cdot N_\mathrm{TC}^{k_{m,e}}, \tag{2.2b}$$

$$\overset{*}{m}_\mathrm{max} = k_{\mathrm{max},0} + k_{\mathrm{max},1} \cdot N_\mathrm{TC}, \tag{2.2c}$$

$$\overset{*}{m} = \overset{*}{m}_\mathrm{s} + (\overset{*}{m}_\mathrm{max} - \overset{*}{m}_\mathrm{s}) \cdot \left[1 - \left(\frac{\Pi}{\Pi_\mathrm{s}}\right)^{c_2}\right]^{\frac{1}{c_1}}. \tag{2.2d}$$

For every turbocharger speed N_TC, these equations define the ellipse-like curve[3]

$$\left(\frac{\Pi}{\Pi_\mathrm{s}}\right)^{c_2} + \left(\frac{\overset{*}{m} - \overset{*}{m}_\mathrm{s}}{\overset{*}{m}_\mathrm{max} - \overset{*}{m}_\mathrm{s}}\right)^{c_1} = 1, \tag{2.3}$$

where the denominators are the semi-axes.

The model is visualised in Fig. 2.6 a). There are two problems that can occur. When no surge model is used, the mass flow is not defined when the surge line is crossed. Already when the surge line is approached, an arbitrarily high sensitivity from pressure ratio to mass flow results. A solution for this problem would be to shift the ellipses such as to exhibit a finite slope at the surge line and to extend this (constant) slope to the ordinate, i.e. $\overset{*}{m} = 0$. The second problem concerns the extrapolation of the turbocharger speed. During the identification of the model parameters by a nonlinear optimisation, only the available data is considered. Thus, at higher rotational speeds, the surge-line mass-flow may be higher than the maximal mass flow. The ellipse is flipped horizontally, leading to an instable behaviour because the mass flow increases with an increasing pressure ratio.

[3] For an ellipse, $c_1 = c_2 = 2$.

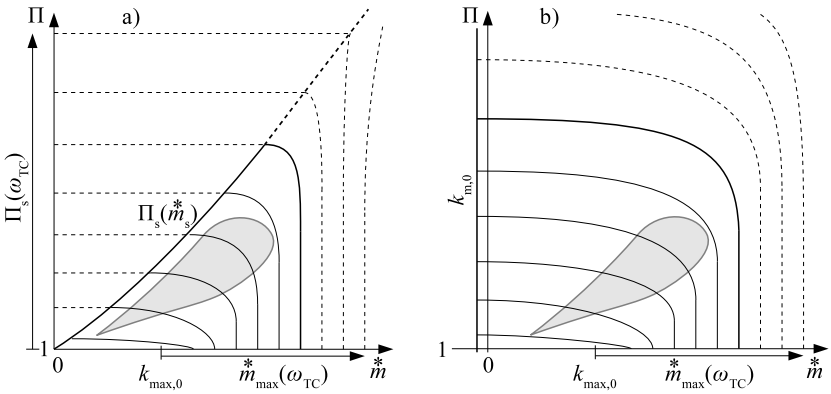

Figure 2.6: a) Original compressor mass-flow model. The extrapolation of the speed lines is critical (dashed speed lines), and the mass flow on the left-hand side of the surge line is ambiguous. The region in which the compressor is operated on the engine is indicated by the shaded area. b) Modified compressor mass-flow model.

The model is modified to prevent the aforementioned critical scenarios during the parameter identification and to ensure a consistent extrapolation of the speed lines. The surge line is replaced by a vertical line with $\dot{m}^*_s$= const. < 0, which is achieved by modifying equations (2.2a) and (2.2b) to

$$\Pi_s = k_{\pi,0} + k_{\pi,1} \cdot N_{TC}^{k_{\pi,e}}, \tag{2.4a}$$

$$\dot{m}^*_s = k_{m,0} \overset{!}{<} 0. \tag{2.4b}$$

To retain sufficient degrees of freedom in the model, a variable shape of the ellipses for different turbocharger speeds is introduced. This effect is usually also observed in compressor maps obtained on compressor

test benches. In the model, the constant parameter c_2 is generalised to

$$c_2 = c_{2,0} + c_{2,1} \cdot N_{\mathrm{TC}} + c_{2,2} \cdot N_{\mathrm{TC}}^2. \tag{2.5}$$

The few parameters of this model can be reliably identified by an automated fitting procedure. The model exhibits good accuracy for all mass-flow ranges and provides sufficient flexibility. The model is visualised in Figure 2.6 b). Various approximations for an ellipse, a more flexible function describing the maximal mass flow, and a variable value of c_1 have been tested, but none of these extensions increases the accuracy of the model.

For the identification of the parameters, a nonlinear search is performed. Both the simplex method in MATLAB's `fminsearch` and the quasi-Newton method in `fminunc` are successful in finding a suitable set of parameters. As the objective function, a combination of the relative and the absolute error terms is used, both represented by the sum of the squares of the respective error. By tuning the weighting between the two error types, more emphasis can be put on the low mass-flow region or on a closer representation of the high-speed range. If a compressor map is available, the speed lines provided therein can be used to enforce a plausible extension of the speed lines towards high mass flows. Initial values for the parameters are obtained by analysing the measurement data available for the engine.

The model is found to be applicable to different engines. Figure 2.7 shows the model identified for engines A and B. The fit shown for engine A is tuned to yield a good approximation in high-speed and high mass-flow regions, and to closely approximate the manufacturer map. Conversely, the model shown for engine B is adjusted such as to yield a good overall fit of the data measured on the engine. For both models, the pressure ratio is calculated using the ambient pressure. If the actual

pressure before the compressor is used, almost identical model fits are achieved.

Compressor enthalpy The enthalpy increase is modelled as described in [183, 139, 132, 137]. Instead of the more common formulation using the isentropic efficiency, the specific enthalpy increase over the compressor is calculated by means of a momentum balance [183]. This momentum balance yields the Euler equation

$$\Delta h = U_2 C_{\theta 2} - U_1 C_{\theta 1}, \tag{2.6}$$

where U_2 is the representative speed of the blade tip. The tangential speed of the gas at inlet $C_{\theta 1}$ is assumed to be zero (axial flow). Under the assumption of no slip, the tangential component of the gas velocity at the outlet is described by the radial component and the back-sweep angle (BSA) β_2 of the blade,

$$C_{\theta 2} = U_2 - C_{r2} \cot(\beta_2). \tag{2.7}$$

The value of C_{r2} is proportional to the mass flow divided by the density at the impeller outlet. In [139], the assumption $\rho_2 = \rho_1$ is found to be reasonable. Inserting (2.7) into (2.6), introducing a constant slip factor, merging all other constant parameters and applying the assumption just mentioned yields

$$\Delta h = k_{\text{slip}} \cdot \left[U_2^2 - k_{\text{BSA}} \cdot U_2 \cdot \frac{\overset{*}{m}}{\rho_1} \right]. \tag{2.8}$$

Rearranging the equations, including the representative radius at the outlet in the corresponding constants and assuming a constant density

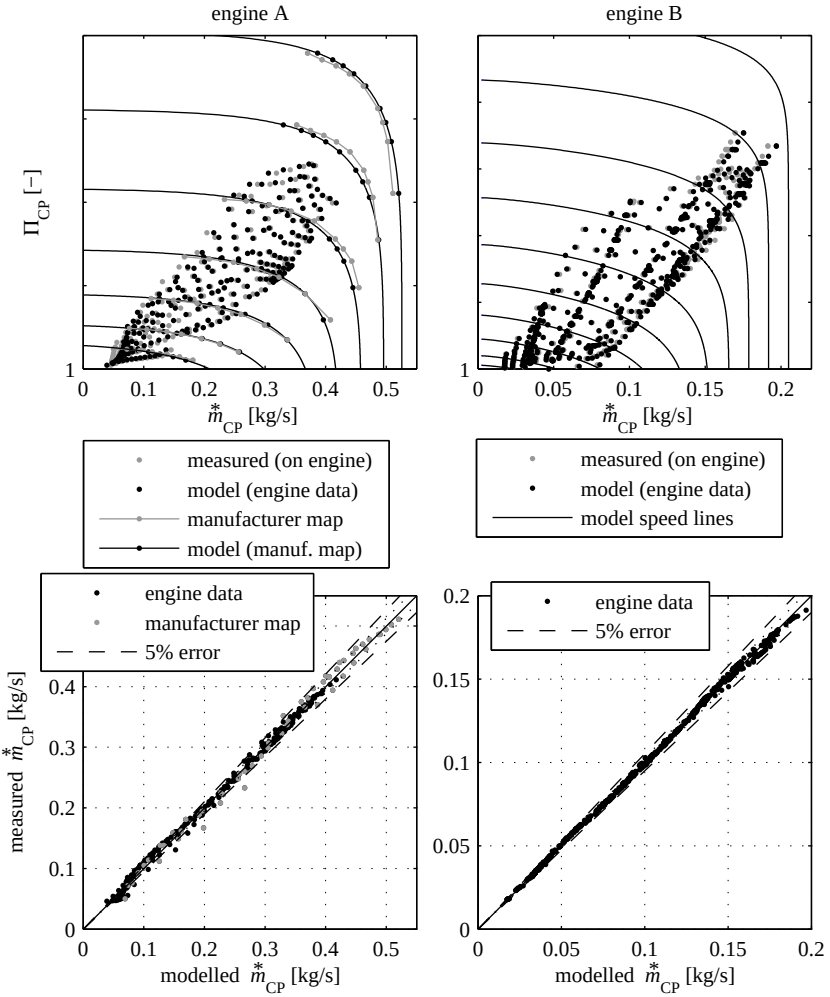

Figure 2.7: Results of the identification of the compressor mass-flow model for engines A and B. To illustrate the generality of the model, the model is tuned for high speeds and mass flows, as well as for a good matching of the manufacturer data for engine A. On engine B, a good overall matching of the data obtained on the engine is aspired. For reasons of confidentiality, the scale for the pressure ratio is hidden.

at the inlet[4] ρ_1 leads to a simple relation for the increase of the specific enthalpy,

$$\Delta h_{\mathrm{CP}} = k_{\mathrm{h},1} \cdot N_{\mathrm{TC}}^2 - k_{h,2} \cdot \overset{*}{m}_{\mathrm{CP}} \cdot N_{\mathrm{TC}}. \tag{2.9}$$

The temperature after the compressor is

$$\vartheta_{\mathrm{CP}} = \vartheta_{\mathrm{amb}} + \frac{\Delta h_{\mathrm{CP}}}{c_{\mathrm{p}}}. \tag{2.10}$$

The two parameters of the model are identified by a linear least-squares (LSQ) regression. If a manufacturer-supplied compressor map is used for the identification, the enthalpy increase has to be calculated from the reference inlet temperature, the pressure ratio and the isentropic efficiency according to

$$\begin{aligned} \Delta h_{\mathrm{CP}} &= c_{\mathrm{p}} \cdot (\vartheta_{\mathrm{CP,out}} - \vartheta_{\mathrm{CP,in}}) \\ &= \frac{1}{\eta_{\mathrm{is}}} \cdot \underbrace{c_{\mathrm{p}} \cdot \vartheta_{\mathrm{CP,in}} \cdot \left(\Pi_{\mathrm{CP}}^{\frac{\kappa-1}{\kappa}} - 1 \right)}_{\Delta h_{\mathrm{is}}}. \end{aligned} \tag{2.11}$$

The model identification for engine A is shown in Fig. 2.8. The underestimated enthalpy increase for low mass flows stems from the heat transfer from the warm casing to the air, which is neglected by the model. No heat transfers occur on the compressor test-bench and thus, the model is accurate for the manufacturer-map data even for low mass flows. The intercooler model can compensate the slightly lower outlet temperature predicted by the compressor-enthalpy model. During the identification of the intercooler model, the modelled compressor-outlet temperature is used instead of the actually measured values.

[4] The assumption of a constant gas density at the inlet is reasonable for single-stage turbocharging only.

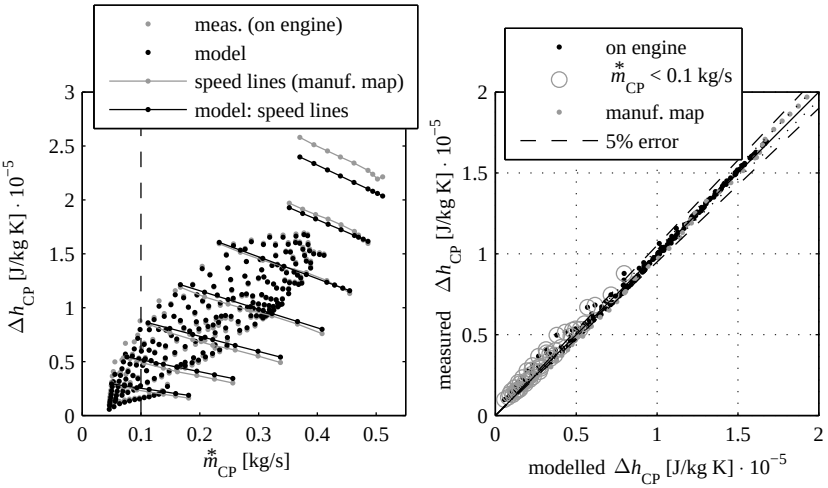

Figure 2.8: Results of the identification of the compressor enthalpy model for engine A. In fact, the constant-speed lines of the manufacturer-supplied map have a constant slope w.r.t. the mass flow. The model inherently prohibits a non-physical compensation of the heat transfers, which is indicated by the trend observed for the low mass-flow measurement data.

Turbine mass-flow For the turbine mass-flow and efficiency, the pressure ratio is defined either as $\Pi_{TB} = p_{EM}/p_{amb}$ or $\Pi_{TB} = p_{EM}/p_4$, according to the model structure chosen. The models described here work well for both versions.

In general, the turbine is treated as an orifice. In contrast to the compressor, it does not actively drive the mass flow by its rotation, but rather the flow is a result of the pressure drop across the turbine. Moreover, the rotation even restricts the flow due to the centrifugal acceleration. Explicitly accounting for this effect leads to a term in N_{TC}^2, which lowers the effective pressure ratio $\Pi_{TB,eff}$ over the turbine [166, p. 131]. It is found that when including such a term, the flow function of

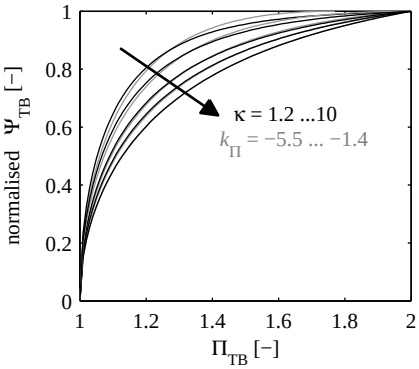

Figure 2.9: Compressible flow function (black) and approximation (2.12) (grey).

the turbine can either be derived from a compressible flow or even be accurately approximated by the simplification

$$\Psi_{\mathrm{TB}} = \sqrt{1 - \Pi_{\mathrm{TB,eff}}^{k_\Pi}}. \tag{2.12}$$

If the centrifugal acceleration is not explicitly accounted for, the parameter identification compensates for this phenomenon by selecting an unphysically high value of $\kappa \approx 4$ or a correspondingly small magnitude of $k_\Pi \approx -2.1$, see Fig. 2.9. This cross-compensation cannot be discerned by steady-state measurements on an engine. When the turbine is operated on an engine, the pressure ratio over the turbine and the turbocharger speed depend on each other. Furthermore, both are mainly defined by the VGT position, which thus is a third dependent variable. During transient operation, however, this coupling of the VGT position, the pressure ratio and the turbocharger speed is broken. The VGT position is a controlled actuator and can be changed almost instantly. The pressure in the exhaust manifold reacts quickly to

a VGT actuation, whereas the turbocharger speed slowly approaches its stationary level due to the inertia of the turbocharger.

This problem of a "flat" input space resulting from the use of stationary measurements for the identification can be partially mitigated when an exhaust flap is present. By varying the opening of this flap, at least the VGT position and the pressure ratio can be decoupled. Using highly transient data to also decouple the turbocharger speed is delicate since heat losses disturb the relations, and obtaining reliable, perfectly time-synchronous measurements of all relevant quantities is difficult. Nevertheless, the validation of the models for transient engine operation shows that the impact of this ambiguity during the identification on the dynamic response of the model is negligible.

In order to apply an LSQ regression, the centrifugal term is assumed not to directly reduce the relevant pressure ratio but is moved outside the square root of the approximate flow function,

$$\overset{*}{\mu}_{\mathrm{TB}} = \left(k_0 + k_1 \cdot u_{\mathrm{VGT}} + k_2 \cdot u_{\mathrm{VGT}}^2 + k_3 \cdot N_{\mathrm{TC}}^2\right) \cdot \sqrt{1 - \Pi_{\mathrm{TB}}^{k_\Pi}}. \tag{2.13a}$$

The normalised mass flow

$$\overset{*}{\mu}_{\mathrm{TB}} = \overset{*}{m}_{\mathrm{TB}} \cdot \frac{\sqrt{\vartheta_{\mathrm{TB,in}}}}{p_{\mathrm{TB,in}}} \tag{2.13b}$$

is used to make the measurement data obtained on the engine comparable to the manufacturer-supplied turbine map. A problem that still renders a direct comparison difficult is the fact that the control signal u_{VGT} is transformed by the actuator chain comprising the power electronics, the electric motor and finally the change of the turbine geometry, e.g. the rotation of the vanes. Therefore, the relation of u_{VGT} to the trim, which is the quantity usually specified in the manufacturer map, is nonlinear and generally unknown.

The best approach is to use the trends observed in the manufacturer map to derive suitable regressors for the model. Subsequently, the data measured on the engine are used to identify the coefficients by an LSQ regression. A selection of regressors that is found to be applicable to different engines is

$$\overset{*}{\mu}_{\mathrm{TB}} = \left(k_0 + k_1 \cdot u_{\mathrm{VGT}} + k_2 \cdot u_{\mathrm{VGT}}^2\right) \cdot \sqrt{\frac{2}{\Pi_{\mathrm{TB}}} \cdot \left(1 - \frac{1}{\Pi_{\mathrm{TB}}}\right)} + k_3 \cdot N_{\mathrm{TC}}. \quad (2.14)$$

In contrast to Eq. (2.13a), it is linear in its parameters. The square root represents an approximation of the flow function for an isothermal orifice, cf. [92, Sec. 2.3.2].

Turbine efficiency The concept of the isentropic efficiency is used for the enthalpy extraction in the turbine,

$$\eta_{\mathrm{TB}} = \frac{\Delta h_{\mathrm{TB}}}{\Delta h_{\mathrm{TB,is}}} = \frac{c_{\mathrm{p}} \cdot (\vartheta_{\mathrm{TB,in}} - \vartheta_{\mathrm{TB,out}})}{c_{\mathrm{p}} \cdot \vartheta_{\mathrm{TB,in}} \cdot \left(1 - \Pi_{\mathrm{TB}}^{\frac{1-\kappa}{\kappa}}\right)}. \quad (2.15)$$

The application of the Euler equation to the turbine does not yield satisfactory results, even if only a single fixed VGT position is considered. One approach to model the efficiency is to define an optimal blade-speed ratio (BSR) for each VGT position, a corresponding maximum efficiency, and a quadratic decrease of the efficiency for deviations from the optimal BSR [92, Sec. 2.3.5]. For the engines at hand, the optimal BSR indeed can be represented by a third-order polynomial in the VGT position. However, the maximum efficiency is a function of the VGT and the turbocharger speed, at least. In the end, too many parameters result, and the model is prone to overfitting.

Again, a reasonable approach is to consider the manufacturer map to derive the main trends, define the regressors accordingly, and use an LSQ regression to match the model to the data obtained on the engine.

During the identification, the enthalpy change over the turbine is matched to the compressor side. In other words, the enthalpy change over the turbine is required to deliver the enthalpy that the compressor transfers to the air,

$$\Delta h_{\mathrm{TB}} \stackrel{!}{=} \frac{\overset{*}{m}_{\mathrm{CP}} \cdot \Delta h_{\mathrm{CP}}}{\overset{*}{m}_{\mathrm{TB}}}. \tag{2.16}$$

This matching ensures an accurate prediction of the turbocharger speed and eliminates the influence of stationary heat losses in the exhaust manifold. These heat losses can be of a substantial magnitude and would thus disturb the notion of the isentropic efficiency if the actually measured outlet temperature was used to define the enthalpy extracted by the turbine. Figure 2.10 illustrates this effect. Only if the temperature after the turbine has to be known, which is the case if the aftertreatment system is included in the model[5], the non-adiabatic efficiency, including the heat losses, has to be estimated as well. Note that the specific heat capacities are omitted in the calculation of the specific enthalpies in (2.16). It is found that using the exact values for the specific heats of the air and the exhaust gas does not improve the matching of the turbine to the compressor.

The following qualitative trends are derived from the manufacturer maps for different VGT turbines.

[5]The temperature after the turbine is one of the inputs to the flow restriction representing the ATS. Furthermore, if e.g. a diesel oxydation catalyst or a selective-catalytic reduction system is modelled, the temperature is critical for the conversion efficiency of these chemical reactors.

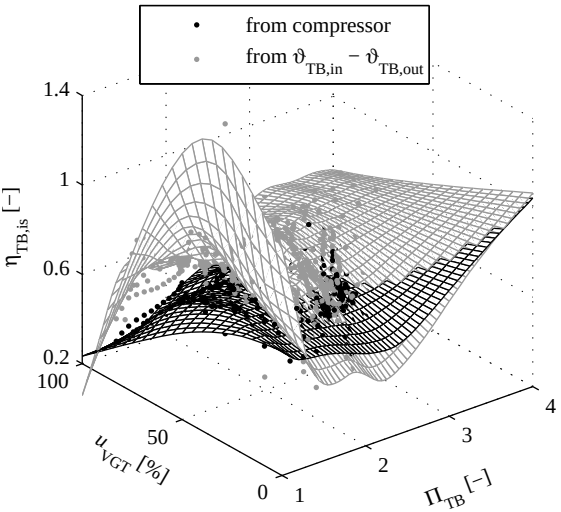

Figure 2.10: Distortion of the turbine efficiency by the (stationary) heat losses. Especially for low pressure ratios (and consequently low mass flows) the effect is obvious. Surfaces are fitted to the data for a better visualisation of the trends.

- The efficiency is approximately a quadratic function of the VGT position.

- The other main characteristic resembles $\sqrt{\Pi_{TB} - 1}$.

- For large pressure ratios Π_{TB}, the efficiency starts to slightly drop again, and the amount of this reduction depends on the VGT position. However, in the region in which the turbine is usually operated on the engine, this effect is hardly perceivable.

Based on these observations and on the evaluation of the model quality for different data sets, the following set of regressors is proposed.

$$\eta_{\mathrm{TB}} = k_{\eta,0} + k_{\eta,1} \cdot u_{\mathrm{VGT}} + k_{\eta,2} \cdot u_{\mathrm{VGT}}^2 + (k_{\eta,3} + k_{\eta,4} \cdot u_{\mathrm{VGT}}) \cdot \sqrt{\Pi_{\mathrm{TB}} - 1} + k_{\eta,5} \cdot N_{\mathrm{TC}}. \quad (2.17)$$

The friction in the turbocharger bearings is not considered explicitly, but rather is lumped into the efficiency. More physics-based models for various turbines can be found in [165, 151].

Intercooler As mentioned already in the introduction to this section, the mass flow through the intercooler is not modelled explicitly. Rather, the small pressure ratio across it is included in the model for the compressor mass-flow. For the temperature difference, two model versions are proposed. Since the intercooler systems of different engines and test-bench setups may differ considerably w.r.t. their construction and the coolant-flow control, this component has to be analysed individually for each engine. The more suitable model version can then be selected and possibly fine-tuned to the engine under consideration.

The first model version is based on a simple stationary energy balance, using the average of the inflow and outflow temperatures as the representative temperature.

$$k_{\mathrm{IC}} \cdot \left(\frac{\vartheta_{\mathrm{CP}} + \vartheta_{\mathrm{IC}}}{2} - \vartheta_{\mathrm{cool}} \right) = \overset{*}{m}_{\mathrm{CP}} \cdot (\vartheta_{\mathrm{CP}} - \vartheta_{\mathrm{IC}}). \quad (2.18a)$$

The efficiency parameter k_{IC} depends on the mass flow,

$$k_{\mathrm{IC}} = k_{\mathrm{IC},1} \cdot \overset{*}{m}_{\mathrm{CP}} + k_{\mathrm{IC},2} \cdot \overset{*}{m}_{\mathrm{CP}}^{\,2}. \quad (2.18b)$$

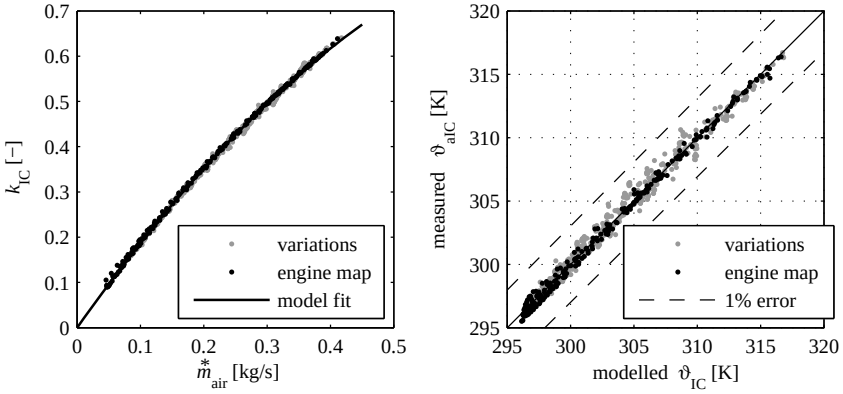

Figure 2.11: The first version of the intercooler model, applied to engine A. The plot to the left shows the perfect values of k_{IC} obtained from the measurement data, as well as the quadratic function (2.18b) fitted to this data. Besides the engine map using the actual calibration, all variations performed are used as data. The plot to the right displays the accuracy of the model. Note that also the temperature *difference* over the intercooler, which ranges from 7 to 160 K, is reproduced with a relative error that never exceeds 2%.

The perfect value of k_{IC} is calculated from the measurement data and an LSQ regression is applied to identify the model parameters $k_{IC,1}$ and $k_{IC,2}$. For the coolant temperature, a constant value is assumed, which can be either tuned by hand or is identified by a nonlinear search. This identification procedure and the model accuracy are illustrated in Fig. 2.11.

The second model version represents the temperature drop over the intercooler directly as a function of the temperature at the inlet and the mass flow. For engine B, the simple relation

$$\vartheta_{CP} - \vartheta_{IC} = k_{IC,0} + k_{IC,1} \cdot \vartheta_{CP} + k_{IC,2} \cdot \dot{m}^*_{CP} + k_{IC,3} \cdot \dot{m}^{*\,2}_{CP} \tag{2.19}$$

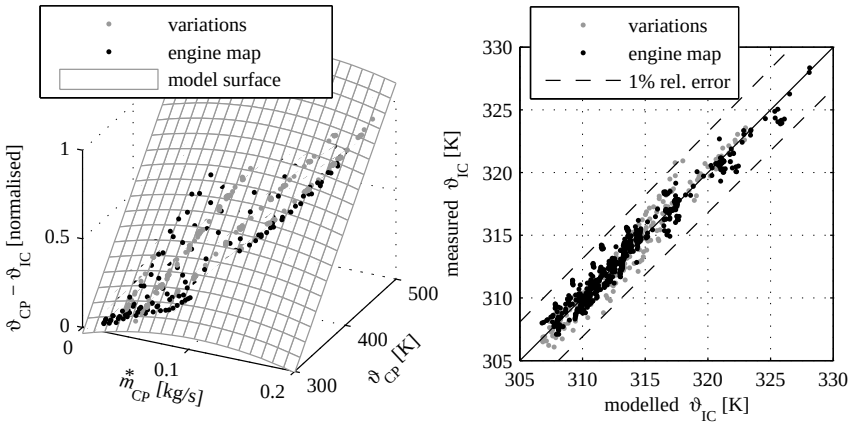

Figure 2.12: The second version of the intercooler model, applied to engine B. As for the case presented in Fig. 2.11, the engine map and all variations performed are used as data. The z-axis of the left-hand plot is normalised for reasons of confidentiality.

is found to yield accurate results, see Fig. 2.12. For other intercooler setups, a different set of regressors may be suitable.

Restrictions The air-flow meter (AFM) and the aftertreatment system (ATS) are modelled as simple flow restrictions. The approximation of the flow function for compressible flows already used for the turbine in Eq. (2.14) is used. Since the pressure ratios never reach critical conditions, no case distinction is necessary. The final model reads

$$\dot{m}^*_i = k_{\mathrm{m},0,i} + k_{\mathrm{m},1,i} \cdot \frac{p_{\mathrm{in},i}}{\sqrt{\vartheta_{\mathrm{in},i}}} \cdot \sqrt{\frac{2}{\Pi_i} \cdot \left(1 - \frac{1}{\Pi_i}\right)}. \tag{2.20}$$

Figure 2.13 reveals that the usual model formulation with $k_{\mathrm{m},0,i} = 0$ exhibits a systematic error. Therefore, the extension to an affine function is used here, i.e. $k_{\mathrm{m},0,i} \neq 0$.

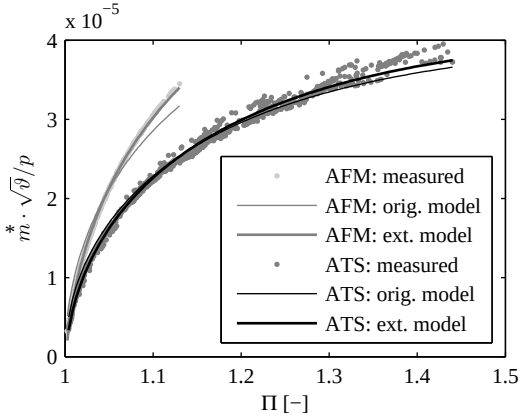

Figure 2.13: Models for the flow restrictions representing the air-flow meter (AFM) and the aftertreatment system (ATS). The original model formulation suffers a systematic error in both cases. For the ATS, it is mainly observable at low pressure ratios.

Cylinder mass-flow The mass flow into the cylinders is defined mainly by the density of the gas in the intake manifold. If sufficient time was available, the intake valve would close exactly at bottom dead centre, and there was no valve overlap, the engine would represent a perfect volumetric pump. The displacement volume of the engine would be pumped through the cylinders every two revolutions, i.e.

$$\overset{*}{m}_{\text{cyl}} = \eta_{\text{vol}} \cdot \frac{p_{\text{IM}}}{R \cdot \vartheta_{\text{IM}}} \cdot \frac{V_{\text{d}}}{2} \cdot \frac{N_{\text{eng}}}{60}, \tag{2.21}$$

with $\eta_{\text{vol}} = 1$.

However, several effects degrade this perfect volumetric efficiency to a value of typically 75 – 95%. A volume of residual gas larger than the volume of the combustion chamber at top dead centre (TDC) may be captured in the cylinder. An exhaust-valve opening before or after

TDC, or flow restrictions leading to a reduction of the pressure gradient may cause such an effect. The same considerations also hold for the flow of fresh gas into the cylinders. Scavenging, which designates the phenomenon that a fraction of the fresh gas flowing into the cylinders directly exits through the exhaust valves, is hardly perceivable in the four-stroke engines considered. While the valve timing is considered to be invariable for a given engine, the flow characteristics are affected by the pressure ratio across the flow and the flow speed, which directly depends on the engine speed.

A model formulation that is able to capture the effects of these phenomena, but still is of a simple nature, is

$$\overset{*}{m}_{\mathrm{cyl}} = \left(k_{\mathrm{vol},0} + k_{\mathrm{vol},1} \cdot N_{\mathrm{eng}} + k_{\mathrm{vol},2} \cdot N_{\mathrm{eng}}^2 + k_{\mathrm{vol},3} \cdot (p_{\mathrm{EM}} - p_{\mathrm{IM}})\right) \cdot \frac{p_{\mathrm{IM}}}{\vartheta_{\mathrm{IM}}}. \quad (2.22)$$

The concept of the volumetric efficiency introduced in Eq. (2.21) is inherently included in this model formulation. For all engines considered, the model reproduces the mass flow into the cylinders with a relative error that slightly exceeds 3% only for some low mass-flow points.

Engine-outlet temperature An enthalpy balance over the cylinders is applied to calculate the temperature of the exhaust gas.

$$c_{\mathrm{p,EO}} \cdot \vartheta_{\mathrm{EO}} \cdot (\overset{*}{m}_{\mathrm{cyl}} + \overset{*}{m}_{\mathrm{fuel}}) - c_{\mathrm{p,IM}} \cdot \vartheta_{\mathrm{IM}} \cdot \overset{*}{m}_{\mathrm{cyl}} = \overset{*}{Q}_{\mathrm{fuel}} - \overset{*}{Q}_{\mathrm{work}} - \overset{*}{Q}_{\mathrm{loss}}. \quad (2.23)$$

The heat flow introduced by the fuel is $\overset{*}{Q}_{\mathrm{fuel}} = H_l \cdot \overset{*}{m}_{\mathrm{fuel}}$. The losses account for the friction and the gas-exhange work defined below in Sec. 2.3,

$$\overset{*}{Q}_{\mathrm{loss}} = (T_{\mathrm{fric}} + T_{\mathrm{gex}}) \cdot \omega_{\mathrm{eng}}. \quad (2.24)$$

In order to enable an identification of the model by an LSQ regression, two assumptions are necessary. First, a constant specific heat of $c_p = 1100\text{J/kg·K}$ is used. The error introduced by this assumption is found to be compensated by the model during the parameter identification. Second, the fuel mass-flow is neglected. It is substantially smaller than the gas flow through the cylinders. At an air-to-fuel ratio of 1.3, the fuel injection amounts to only 5% of the air mass-flow into the cylinders. This percentage further decreases for higher air-to-fuel ratios or in the presence of EGR. As for the assumption of a constant specific heat, the model is able to compensate for this slight simplification.

As a last step, the right-hand-side of Eq. (2.23) is rearranged. The heat that remains in the exhaust gas is a fraction of the heat introduced by the fuel that is not consumed by the losses. The simplified enthalpy balance thus reads

$$c_p \cdot \overset{*}{m}_{cyl} \cdot (\vartheta_{EO} - \vartheta_{IM}) = x_\vartheta \cdot \left(\overset{*}{Q}_{fuel} - \overset{*}{Q}_{loss} \right). \tag{2.25}$$

The error in the difference between the temperatures of the exhaust gas and the gas in the intake manifold has to be minimised. By reformulating (2.25) as

$$\vartheta_{EO} - \vartheta_{IM} = \frac{\overset{*}{Q}_{fuel} - \overset{*}{Q}_{loss}}{c_p \cdot \overset{*}{m}_{cyl}} \cdot x_\vartheta, \tag{2.26}$$

the coefficients of the regressors defining x_ϑ can by identified by an LSQ regression that minimises the squared error of the temperature difference.

The fraction x_ϑ becomes smaller for a higher combustion efficiency. Therefore, the control inputs of the combustion have to be included in the regressors. Another fraction of the heat that does not show in

the exhaust gas stems from the heat losses to the combustion-chamber walls. The engine speed defines the time available for the heat transfer, and the cylinder mass forms the "cushion" between the combustion zone and the walls. A set of regressors that suits all engines considered is

$$x_\vartheta = k_{\mathrm{x},0} + k_{\mathrm{x},1} \cdot N_{\mathrm{eng}} + k_{\mathrm{x},2} \cdot N_{\mathrm{eng}}^2 + k_{\mathrm{x},3} \cdot \overset{*}{m}_{\mathrm{cyl}} + k_{\mathrm{x},4} \cdot \overset{*}{m}_{\mathrm{cyl}}^{\,2} + k_{\mathrm{x},5} \cdot \varphi_{\mathrm{SOI}} + k_{\mathrm{x},6} \cdot p_{\mathrm{rail}}. \quad (2.27)$$

The maximum magnitude of the relative error is below 5% for the full engine maps and all variations.

Dynamic elements The dynamics of the air path are described by balance equations. A storage quantity such as mass or energy is changed by corresponding flows into and out of the storage. The storage level is represented by a state variable, which does not necessarily have to be chosen as the physically stored quantity. For example, the mass and the internal energy stored in a volume are represented by the pressure and the temperature inside the volume. The flows between any two storage devices are driven by the difference of their state variables.

The models presented thus far serve to calculate the flows from the state variables. In this paragraph, the equations for the storage elements are derived as functions of those flows. For the turbocharger speed, a rotational momentum balance yields

$$\Theta_{\mathrm{TC}} \cdot \frac{d\omega_{\mathrm{TC}}}{dt} = T_{\mathrm{TB}} - T_{\mathrm{CP}} = \frac{\overset{*}{m}_{\mathrm{TB}} \Delta h_{\mathrm{TB}} - \overset{*}{m}_{\mathrm{CP}} \Delta h_{\mathrm{CP}}}{\omega_{\mathrm{TC}}}. \quad (2.28a)$$

Recall that friction is accounted for by the model for the turbine efficiency.

In the models for the engines without EGR, the temperatures inside the intake and exhaust manifolds are assumed to be defined by the respective inflowing gas. Therefore, only the mass balances remain,

$$\frac{dp_{\mathrm{IM}}}{dt} = \frac{R \cdot \vartheta_{\mathrm{IM}}}{V_{\mathrm{IM}}} \cdot (\overset{*}{m}_{\mathrm{CP}} - \overset{*}{m}_{\mathrm{cyl}}), \tag{2.28b}$$

$$\frac{dp_{\mathrm{EM}}}{dt} = \frac{R \cdot \vartheta_{\mathrm{EM}}}{V_{\mathrm{EM}}} \cdot (\overset{*}{m}_{\mathrm{cyl}} + \overset{*}{m}_{\mathrm{fuel}} - \overset{*}{m}_{\mathrm{TB}}). \tag{2.28c}$$

If the thermal models of the intake and the exhaust manifolds are not included, the temperatures of the gases in the manifolds are defined by the flows entering them, i.e. $\vartheta_{\mathrm{IM}} = \vartheta_{\mathrm{IC}}$ and $\vartheta_{\mathrm{EM}} = \vartheta_{\mathrm{EO}}$. For the two additional receivers in the model version including the AFM and ATS restrictions, equations similar to (2.28b) apply.

2.2.2 Exhaust-Gas Recirculation

One version of engine B features an exhaust-gas recirculation (EGR) system. This system consists of a pipe with a small diameter that connects the exhaust and the intake manifolds. A valve is used to control the mass flow, and a cooler ensures that the recirculated exhaust gas does not substantially heat up the mixture in the intake manifold. The goal of recirculating exhaust gas into the intake manifold is to reduce the NO_x emissions of the engine. The burnt gas in the intake manifold increases the fraction of inert gas in the cylinders. This dilution has the effect that a higher gas mass participates in the combustion and therefore, the heat of the combustion is more widely distributed. The resulting lower temperatures in the regions close to the flame lead to a reduction of the thermal NO_x formation, which is highly sensitive towards the temperature.

The negative effect of EGR is that the combustion becomes less efficient due to the lower temperatures. Furthermore, the reduced

oxygen concentration leads to a slightly slower combustion. Both effects decrease the fuel economy. However, the tradeoff between the NO_x emissions and the fuel consumption is generally better than with measures acting on the combustion, such as the start of injection. On the other hand, the lower oxygen availability and the generally worse combustion increase the soot emissions.

The following assumptions are made to keep the dynamics of the air-path model simple. They originate from the presumption of a plug flow through the EGR piping.

- No transport delay of the mass through the EGR system. In a plug flow, the flow through the EGR valve instantly "pushes" the same flow into the intake manifold.
- No gas dynamics from the exhaust ports to the EGR valve. In accordance with the plug-flow presumption, no mixing or diffusion takes place.
- No transport delay of the gas composition through the EGR system. The flow entering the intake manifold is assumed to have the composition of the exhaust gas leaving the exhaust ports at the same time.
- Perfectly mixed gas in the intake manifold. This assumption is common to all mean-value modelling approaches.
- The EGR cooler is installed as close to the valve that the two parts can be represented by a single flow restriction with a variable opening area.

The gas in the intake manifold is subdivided into two species. Two different choices of these species are common. The first one considers burnt gas and air. Burnt gas thereby denotes the mixture that actually

participated in the (stoichiometric) combustion, and thus no oxygen is left in it. The burnt-gas fraction in the exhaust gas is

$$x_{\mathrm{BG,EM}} = \frac{x_{\mathrm{BG,IM}} \cdot \overset{*}{m}_{\mathrm{cyl}} + (1+\sigma_0) \cdot \overset{*}{m}_{\mathrm{fuel}}}{\overset{*}{m}_{\mathrm{cyl}} + \overset{*}{m}_{\mathrm{fuel}}}, \tag{2.29}$$

where σ_0 is the stoichiometric air-to-fuel ratio.

The second option is to consider the exhaust gas and fresh air as the two gas species. Obviously, the fraction of exhaust gas is 1 in the exhaust manifold. Apart from this difference, the derivation of the balance equations for the intake manifold is the same for the two cases. In the following, j denotes either burnt gas or exhaust gas. The two mass-balance equations in the intake manifold read

$$\begin{aligned} \frac{dm_{\mathrm{IM,air}}}{dt} &= \overset{*}{m}_{\mathrm{CP,air}} + \overset{*}{m}_{\mathrm{EGR,air}} - \overset{*}{m}_{\mathrm{cyl,air}} \\ &= \overset{*}{m}_{\mathrm{CP}} + \overset{*}{m}_{\mathrm{EGR}} \cdot (1 - x_{j,\mathrm{EM}}) - \overset{*}{m}_{\mathrm{cyl}} \cdot (1 - x_{j,\mathrm{IM}}), \end{aligned} \tag{2.30a}$$

$$\frac{dm_{\mathrm{IM},j}}{dt} = \overset{*}{m}_{\mathrm{EGR}} \cdot x_{j,\mathrm{EM}} - \overset{*}{m}_{\mathrm{cyl}} \cdot x_{j,\mathrm{IM}}. \tag{2.30b}$$

In the energy balance, constant and uniform specific heats are assumed,

$$\begin{aligned} \frac{d}{dt}(c_{\mathrm{v}} \cdot m_{\mathrm{IM}} \cdot \vartheta_{\mathrm{IM}}) &= c_{\mathrm{v}} \cdot \left(m_{\mathrm{IM}} \cdot \frac{d\vartheta_{\mathrm{IM}}}{dt} + \vartheta_{\mathrm{IM}} \cdot \frac{dm_{\mathrm{IM}}}{dt} \right) \\ &= c_{\mathrm{p}} \cdot \left(\overset{*}{m}_{\mathrm{CP}} \cdot \vartheta_{\mathrm{IC}} + \overset{*}{m}_{\mathrm{EGR}} \cdot \vartheta_{\mathrm{EGR}} - \overset{*}{m}_{\mathrm{cyl}} \cdot \vartheta_{\mathrm{IM}} \right), \end{aligned} \tag{2.30c}$$

where $m_{\mathrm{IM}} = m_{\mathrm{IM,air}} + m_{\mathrm{IM},j}$. Application of the ideal gas law enables the solution of the equation system (2.30) for the pressure, the temperature

and the mass fraction of the chosen species in the intake manifold,

$$\frac{dp_{\mathrm{IM}}}{dt} = \frac{R\cdot\kappa}{V_{\mathrm{IM}}}\cdot\left(\overset{*}{m}_{\mathrm{CP}}\cdot\vartheta_{\mathrm{IC}} + \overset{*}{m}_{\mathrm{EGR}}\cdot\vartheta_{\mathrm{EGR}} - \overset{*}{m}_{\mathrm{cyl}}\cdot\vartheta_{\mathrm{IM}}\right), \tag{2.31a}$$

$$\frac{d\vartheta_{\mathrm{IM}}}{dt} = \frac{R\cdot\vartheta_{\mathrm{IM}}}{p_{\mathrm{IM}}\cdot V_{\mathrm{IM}}\cdot c_{\mathrm{v}}}\cdot\left[c_{\mathrm{p}}\cdot\left(\overset{*}{m}_{\mathrm{CP}}\cdot\vartheta_{\mathrm{IC}} + \overset{*}{m}_{\mathrm{EGR}}\cdot\vartheta_{\mathrm{EGR}} - \overset{*}{m}_{\mathrm{cyl}}\cdot\vartheta_{\mathrm{IM}}\right)\right.$$
$$\left. - c_{\mathrm{v}}\cdot\vartheta_{\mathrm{IM}}\cdot\left(\overset{*}{m}_{\mathrm{CP}} + \overset{*}{m}_{\mathrm{EGR}} - \overset{*}{m}_{\mathrm{cyl}}\right)\right], \tag{2.31b}$$

$$\frac{dx_{j,\mathrm{IM}}}{dt} = \frac{R\cdot\vartheta_{\mathrm{IM}}}{p_{\mathrm{IM}}\cdot V_{\mathrm{IM}}}\cdot\left(\overset{*}{m}_{\mathrm{EGR}}\cdot(x_{j,\mathrm{EM}} - x_{j,\mathrm{IM}}) - x_{j,\mathrm{IM}}\cdot\overset{*}{m}_{\mathrm{CP}}\right). \tag{2.31c}$$

The advantage of the burnt-gas fraction over the exhaust-gas fraction is the fact that the composition of the burnt gas depends only on the composition of the air. Therefore, it is changed only by the ambient conditions, which can be assumed to be constant, or at least to change significantly more slowly than the EGR dynamics. For transient operation, where the air-to-fuel ratio can change rapidly, this choice of species thus enables a more accurate prediction of the composition of the mixture in the intake manifold. For this reason, the burnt-gas fraction is used throughout this work.

The composition of the burnt gas ξ_{BG} is pre-calculated using the air composition obtained from the ambient conditions, and by assuming stoichiometric combustion, see Appendix B. The calculation of the cylinder-charge composition before and after the combustion remains the same as for an engine without EGR, but the mixture flowing into the cylinders is not air but has the composition

$$\xi_{\mathrm{IM}} = (1 - x_{\mathrm{BG,IM}})\cdot\xi_{\mathrm{air}} + x_{\mathrm{BG,IM}}\cdot\xi_{\mathrm{BG}}. \tag{2.32}$$

EGR valve The approximate flow function already used for the AFM and the ATS in Eq. (2.20) is applied, i.e.

$$\Psi(\Pi_{\mathrm{EGR}}) = \sqrt{\frac{2}{\Pi_{\mathrm{EGR}}} \cdot \left(1 - \frac{1}{\Pi_{\mathrm{EGR}}}\right)}, \tag{2.33}$$

with $\Pi_{\mathrm{EGR}} = p_{\mathrm{EM}} / p_{\mathrm{IM}}$. To identify the opening area corresponding to a given EGR-valve position u_{EGR}, the perfect values for the effective opening area are calculated from the measurement data by

$$(c_{\mathrm{d}} \cdot A)_{\mathrm{EGR}} = \overset{*}{m}_{\mathrm{EGR}} \cdot \frac{\sqrt{R \cdot \vartheta_{\mathrm{EO}}}}{p_{\mathrm{EM}}} \cdot \frac{1}{\Psi(\Pi_{\mathrm{EGR}})}. \tag{2.34}$$

The resulting values are plotted against u_{EGR} to identify the relation between the control signal and the effective opening area. The left-hand plot of Fig. 2.14 shows this data (“original formulation”).

The model quality obtained when fitting the curve

$$(c_{\mathrm{d}} \cdot A)_{\mathrm{EGR}} = k_{\mathrm{EGR},1} \cdot (u_{\mathrm{EGR}} - u_{\mathrm{EGR},0}) + k_{\mathrm{EGR},2} \cdot (u_{\mathrm{EGR}} - u_{\mathrm{EGR},0})^{k_{\mathrm{EGR},e}} \tag{2.35}$$

to this data is shown in the right-hand plot. Obviously, the prediction is poor at low pressure ratios. The EGR system has no reed valves installed, but some other fluid-dynamic effects seem to alter the behaviour of the flow restriction for different pressure ratios. A thorough investigation of this phenomenon could not be performed as of yet. Partially, this mismatch could be caused by the estimation of the EGR mass-flow by the volumetric efficiency of the engine instead of the measurement of a gas-species concentration in the intake manifold, see Sec. C.2.1 in the Appendix.

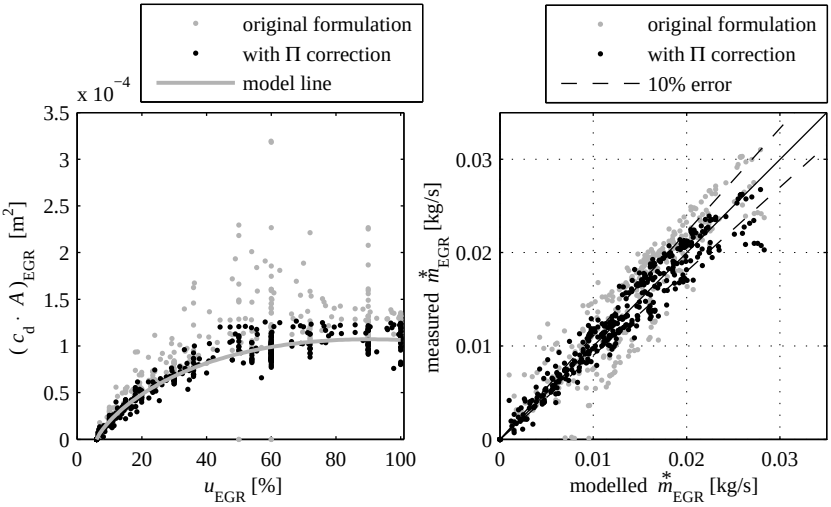

Figure 2.14: Model of the EGR valve. The introduction of a correction factor for the pressure ratio enables the reproduction of the qualitative behaviour of the valve.

An intermediate workaround for this problem is the introduction of a correction factor for the pressure ratio,

$$\tilde{\Pi}_{\mathrm{EGR}} := c_{\Pi} \cdot \Pi_{\mathrm{EGR}}. \tag{2.36}$$

This factor is identified by an outer nonlinear optimisation along with $k_{\mathrm{EGR},e}$, while the parameters $k_{\mathrm{EGR},1}$ and $k_{\mathrm{EGR},2}$ are identified by an LSQ regression inside the nonlinear objective function. The resulting correction factor is $c_{\Pi} = 1.0415$ and the result, as well as the model fit for $(c_{\mathrm{d}} \cdot A)_{\mathrm{EGR}}$, are shown in Fig. 2.14.

EGR cooler The EGR cooler is modelled according to the second version of the intercooler model, cf. Eq. (2.19). A slightly different set of

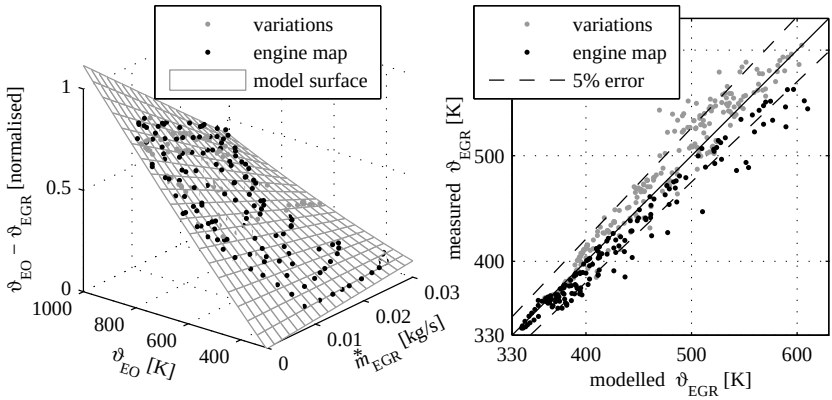

Figure 2.15: Model of the EGR cooler. A slight systematic difference between the engine map and the variation data is present. The z-axis of the left-hand plot is normalised for reasons of confidentiality.

regressors is required to achieve a satisfactory match between model and data,

$$\vartheta_{EO} - \vartheta_{EGR} = k_{EGC,0} + k_{EGC,1} \cdot \vartheta_{EO} + k_{EGC,2} \cdot \vartheta_{EO}^2 + k_{EGC,3} \cdot \dot{m}^*_{EGR} + k_{EGC,4} \cdot \vartheta_{EO} \cdot \dot{m}^*_{EGR}. \quad (2.37)$$

The results are presented in Fig. 2.15. The engine map with EGR, as well as the variations of the EGR rate are used during the identification. A slight systematic difference in the data sets of the engine map and the EGR variations can be observed. The reason for this mismatch is not known. Possibly, a different setpoint for the outlet temperature was used for the two measurement sets. The LSQ regression chooses the model parameters such that the model surface is between the two divergent trends, but the larger number of measurement points introduces a bias of the fit towards the variation data.

Exhaust flap In engine B, an additional throttle is installed after the turbine. This "exhaust flap" (EF) may be used to increase the exhaust-manifold pressure to enable higher EGR rates. Figure 2.16 shows the effect of an EF variation in one variation point. The VGT is actuated such as to maintain the boost-pressure reference. Since the efficiency of the turbocharger does not change significantly, not only the boost pressure but also the turbocharger speed remains constant. In contrast, the model without the EF considers only the influence of the more closed VGT, which leads to a speedup of the turbocharger. The model results depicted in Fig. 2.16 are calculated by running a simulation of the model with constant control inputs until steady state is reached.

The EF is mounted so close to the turbine outlet that there is no space remaining for any sensors between the two components. Therefore, the pressure and the temperature before the flap are not available for the identification of this submodel, and an indirect identification has to be performed.

Directly inverting the model for the mass flow through the turbine is not feasible due to the vanishing slope of the flow function at high mass flows. Furthermore, the temperature after the turbine is hard to be modelled accurately, see paragraph *Turbine efficiency* on page 37. Therefore, the following approach is chosen here.

- Use the model version that includes the restrictions representing the AFM and the ATS.
- The effect of the EF is modelled as a reduction of the effective opening area of the ATS restriction, i.e.

$$k_{\mathrm{m,1,ATS}} \rightarrow x_{\mathrm{EF}} \cdot k_{\mathrm{m,1,ATS}} \tag{2.38}$$

  in Eq. (2.20). In Fig. 2.16, the results from the model with half the ATS opening area are shown. The trends of the measurements

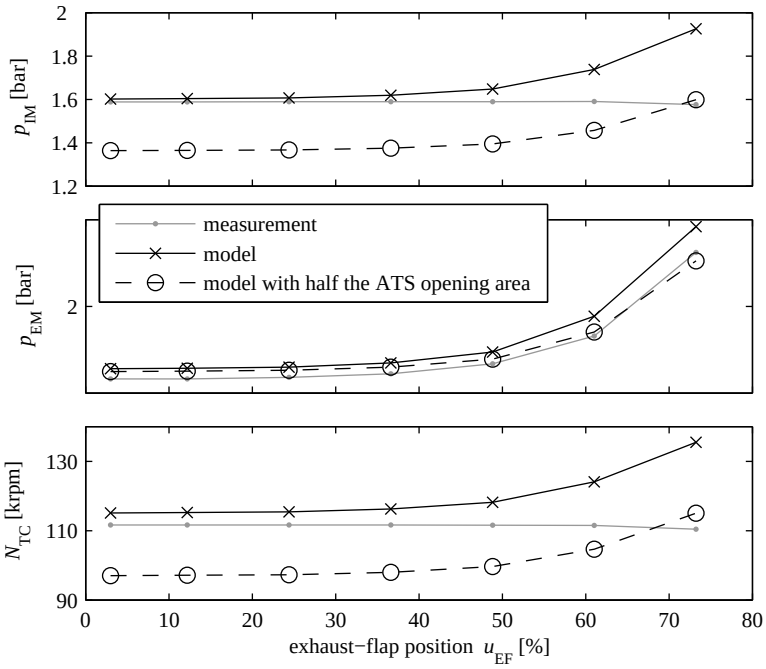

Figure 2.16: Effect of the exhaust flap on the exhaust pressure, the boost pressure, and the turbocharger speed. Operating point: 1900 rpm, 60% load. For reasons of confidentiality, the scale for the exhaust pressure is hidden.

are predicted correctly, i.e. a similar increase of the pressure in the exhaust manifold is retained while the boost pressure and the turbocharger speed are almost unchanged as compared to the reference point with a fully open EF. In this modelling approach, the state variable p_4 always represents the pressure between the turbine and the EF. The pressure between the EF and the ATS is not a state variable of the model.

- Identify the factor for the opening-area reduction individually for all measurement points by a nonlinear search[6]. The objective function is the sum of the squared errors in the predicted change of the pressures in the intake and exhaust manifolds. The turbocharger speed is not explicitly included since it correlates with the intake-manifold pressure and is not of primary interest. The errors are normalised by the measured reference value to ensure an equal weighting,

$$e_{\mathrm{EF}} := \left(\frac{[\hat{p}_{\mathrm{IM}}(u_{\mathrm{EF}}) - \hat{p}_{\mathrm{IM}}(0)] - [p_{\mathrm{IM}}(u_{\mathrm{EF}}) - p_{\mathrm{IM}}(0)]}{p_{\mathrm{IM}}(0)} \right)^2 + \left(\frac{[\hat{p}_{\mathrm{EM}}(u_{\mathrm{EF}}) - \hat{p}_{\mathrm{EM}}(0)] - [p_{\mathrm{EM}}(u_{\mathrm{EF}}) - p_{\mathrm{EM}}(0)]}{p_{\mathrm{EM}}(0)} \right)^2 . \tag{2.39}$$

  Here, the hat designates the model prediction.

  It is important to set the function-value tolerance of the optimisation one magnitude higher than the tolerance of the ODE solver used to simulate the model. Similarly, if a gradient-based optimisation algorithm is chosen, the perturbation for the finite differences has to be chosen sufficiently large to provide consistent derivatives.

- The result of this procedure are measurement-point individual reduction factors for the ATS opening area. They are shown in Fig. 2.17.

[6] The MATLAB-function *fminsearch* is found to be more reliable for this task, although it requires more iterations and function evaluations than the quasi-Newton method implemented in *fminunc*.

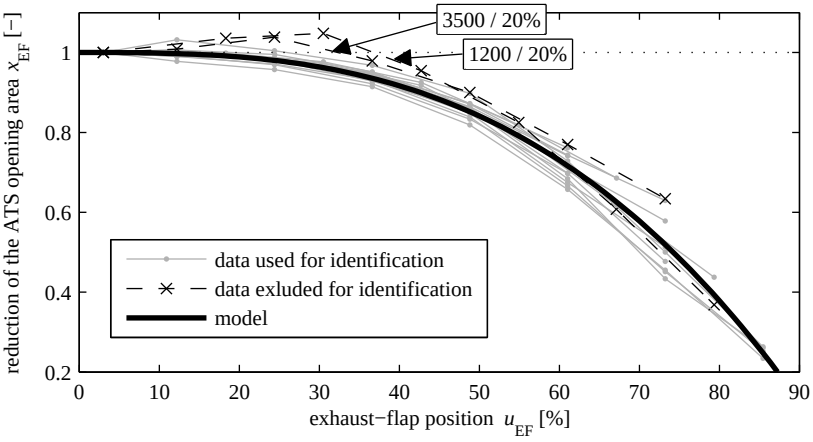

Figure 2.17: Identification of the exhaust-flap model. The two extreme operating points are exluded from the identification data in order to avoid disturbing the resulting parameter values.

- The relation between the control signal and the opening area is typical for a rotational throttle. A suitable model is

$$x_{\mathrm{EF}} = 1 - k_{\mathrm{EF},1} \cdot u_{\mathrm{EF}}^{k_{\mathrm{EF},2}}. \tag{2.40}$$

  The factor $k_{\mathrm{EF},1}$ is identified by an LSQ regression, and the exponent $k_{\mathrm{EF},2} = 2.9$ is tuned by hand. The resulting model line is shown in Fig. 2.17.

Figure 2.18 presents the validation of the overall model including the EF. The turbocharger speed and the intake-manifold pressure are predicted accurately. The accuracy of the exhaust-manifold pressure is similar to that obtained with the model without the EF. Only the most extreme variations, which are not plausible during normal operation of the engine, exhibit relative errors that exceed 5%.

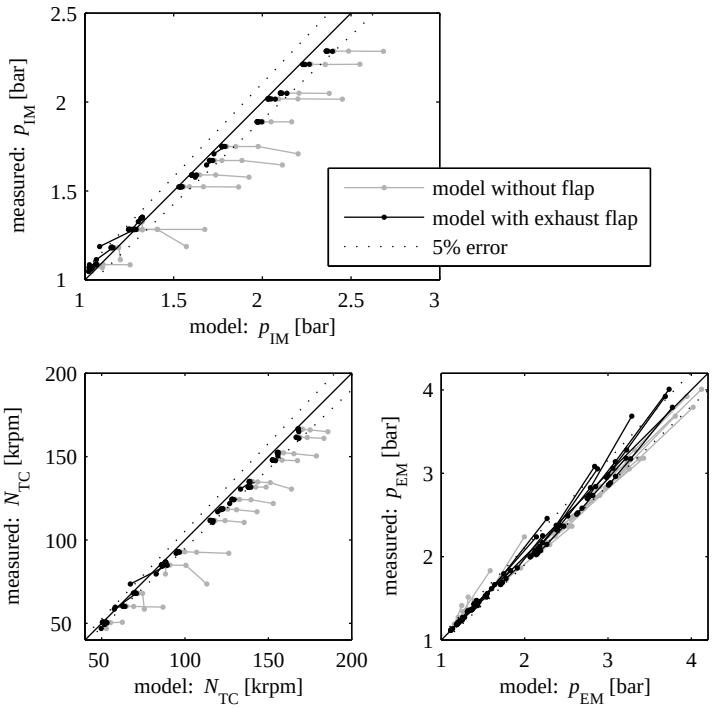

Figure 2.18: Validation of the exhaust-flap model.

2.2.3 Thermal model

In this section, an optimisation-oriented thermal model for diesel engines is developed. The focus is on the transient effects since stationary heat losses are accounted for inherently by the use of stationary measurements for the identification of the relevant components, e.g. the intercooler and the engine-out temperature. The main purpose of the model is to improve the dynamic response of the turbocharger, since the enthalpy available to the turbine is overestimated if no transient heat losses in the exhaust manifold are considered. The second goal

is to provide an accurate prediction of the temperature in the intake manifold, which is critical for the NO_x emissions, see Sec. 2.5.5 below.

For the development of the model, specific measurement data obtained on engine E is used. During these measurements, the gas temperatures are recorded by fast 0.5mm thermocouples, and the surface temperatures of the exhaust manifold and of the turbocharger casing are measured at different locations. The placement of the surface-temperature sensors is shown in Fig. 2.1. Besides a full stationary engine map, three types of transient operation are recorded, namely load steps at three engine-speed levels as well as steps in the VGT position and the SOI on a 3×3 load-speed grid. The load steps have the most significant impact on the exhaust-gas temperature. Varying the VGT mainly changes the mass flow through the engine, whereas the SOI exclusively affects the engine-out temperature. The latter two types of measurements are used to isolate certain influences and to validate assumptions.

Analysis of the surface temperatures Figure 2.19 shows the surface temperatures during the load steps at various engine speeds. All the parts of the exhaust system are heated up similarly fast, except for the outermost part of the collector (at cylinder 6). The latter is slightly cooler since it is only heated by the exhaust gas of one cylinder and is most exposed to the ambient air. The average of the temperatures at all other locations is plotted in black. Since the dynamic behaviour is uniform, the full exhaust manifold and the turbine casing are lumped into a single thermal capacity, henceforth termed the “casing”.

The stationary casing temperature can be estimated from the gas temperature at the confluence point, where the exhaust-gas temperature is usually measured, and the exhaust-gas mass-flow. Due to heat losses to the ambient air, the casing never fully heats up to the temper-

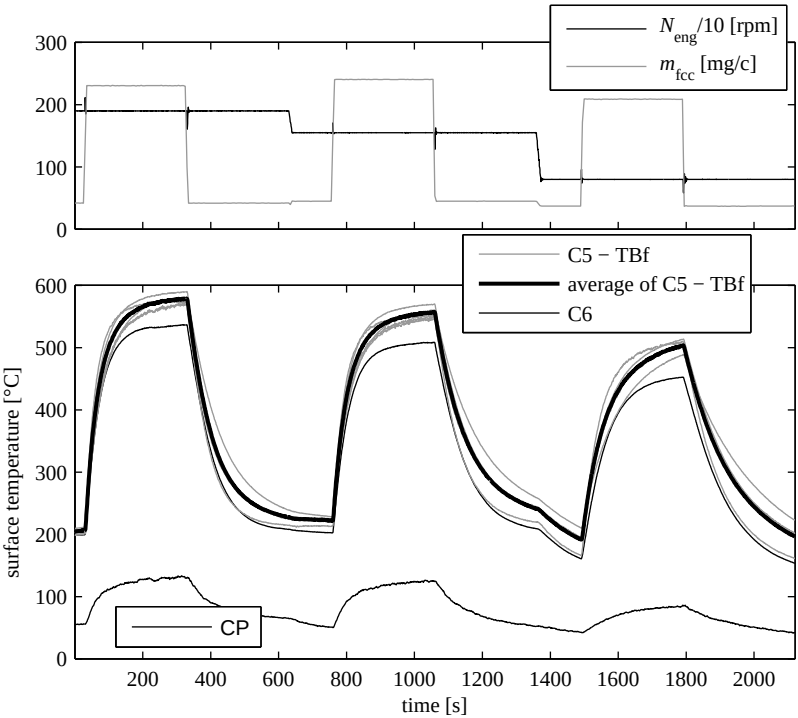

Figure 2.19: Surface temperatures at all measurement points. The compressor casing is only slightly heated up. The exhaust-manifold walls and the turbine casing are heated up rather uniformly, with the exception of the outermost part of the collector.

ature of the exhaust gas. The mass-flow defines the heat-transfer rate and, together with the gas temperature, the enthalpy flow. The simple model

$$\bar{\vartheta}_\mathrm{c} = k_0 + \bar{\vartheta}_\mathrm{EO} \cdot \left(k_1 + k_2 \cdot \overset{*}{m}_\mathrm{exh} + k_3 \cdot \overset{*}{m}{}^2_\mathrm{exh} \right) \tag{2.41}$$

is able to reproduce the casing temperature with a relative error of less than 2%. The bar designates a stationary value. The quadratic term in the mass flow only slightly improves the model quality and will be omitted later on. The parameters $k_{0\ldots3}$ are identified as 31.9, 0.792, 0.448 and -0.426. The strong correlation between $\bar{\vartheta}_\mathrm{EO}$ and $\bar{\vartheta}_\mathrm{c}$ is indicated by the value of 79.2% of the corresponding coefficient.

Dynamics of the casing temperature The identification of the thermal capacity of the casing and of the heat-transfer coefficients is not unambiguous. If all heat transfers are scaled by the same factor, the same effect is achieved as if the thermal capacity was changed. Therefore, a more empirically motivated approach is chosen. A pure low-pass element is applied to the stationary casing temperature,

$$\frac{d\vartheta_\mathrm{c}}{dt} = \frac{1}{\tau_\mathrm{c}} \cdot (\bar{\vartheta}_\mathrm{c} - \vartheta_\mathrm{c}). \tag{2.42}$$

The characteristic time τ_c depends on the mass flow, which defines the rate of the heat transfer. However, the order of magnitude of the characteristic time is defined by the thermal inertia of the casing.

To identify a model for the characteristic time, the load-step measurement is subdivided into small segments. On each of these segments, a nonlinear optimisation of the characteristic time is performed. The integral of the squared difference between the modelled and the measured casing temperature is used as the objective function. The result-

ing data indicates that the characteristic time can be approximated as an affine function of the mass flow,

$$\tau_c = \tau_{c,0} + \tau_{c,1} \cdot \overset{*}{m}_{exh}, \tag{2.43}$$

with $\tau_{c,1} < 0$.

The VGT and the SOI variations are used to validate this model. Due to the lack of an accurate exhaust-gas temperature model for this engine, the measured temperature is used as the "static" gas temperature $\bar{\vartheta}_{EO}$ in Eq. (2.41). Since the transient heat transfers are small for the VGT and the SOI variations, this procedure allows a qualitative assessment of the model. Figure 2.20 shows the results for one operating point. The model is able to predict the qualitative behaviour of the casing temperature also for the VGT and the SOI variations, provided that accurate models for the exhaust-gas temperature and for the mass-flow are available. For the variations at the other operating points, an even closer match between the measurement data and the model output is observed.

Gas temperature The difference between the temperatures of the gas and the casing drives heat transfers which change the exhaust-gas temperature. Due to the relatively small temperature differences and the high flow rates, convective heat transfer is assumed to be dominant [92, Sec. 2.6]. The heat transfer from the exhaust gas to the manifold walls therefore is

$$\overset{*}{Q}_w = \underbrace{\alpha_w \cdot A_w}_{=:k_w} \cdot \underbrace{(\bar{\vartheta}_{EO} - \vartheta_c)}_{\Delta\vartheta_w}, \tag{2.44}$$

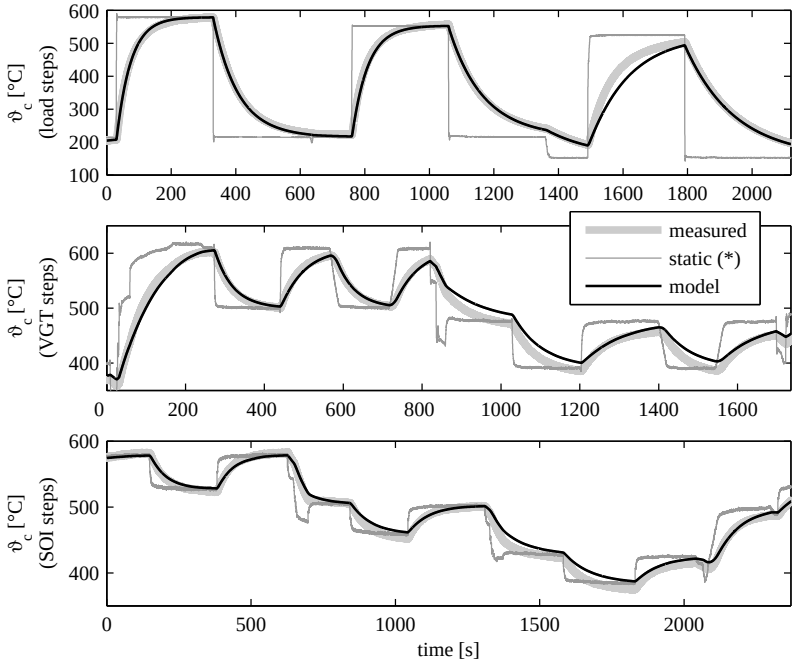

Figure 2.20: Validation of the casing-temperature model. (*) The measured exhaust-gas temperature is used for the VGT and the SOI steps, which are performed at an engine speed of 1600 rpm and for three different loads.

resulting in a temperature change of the gas,

$$\bar{\vartheta}_{\mathrm{EO}} - \vartheta_{\mathrm{EO}} = \Delta\vartheta_{\mathrm{EO}} = \frac{\overset{*}{Q}_{\mathrm{w}}}{c_{\mathrm{p}} \cdot \overset{*}{m}_{\mathrm{exh}}}. \tag{2.45}$$

For forced convection, the heat-transfer coefficient k_{w} depends quadratically on the mass flow. By further assuming a constant c_{p}, Eqs. (2.44)

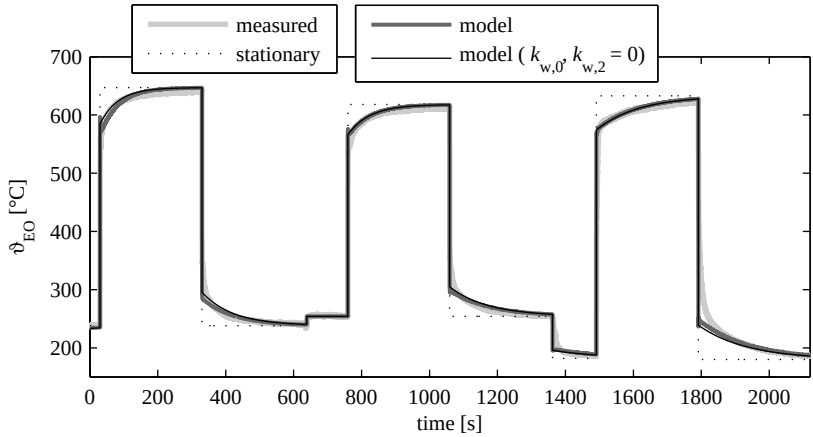

Figure 2.21: Identification of the model for the temperature reduction of the exhaust gas due to transient heat losses. The model with only one parameter is only negligibly less accurate than the full model.

and (2.45) combine to

$$\Delta\vartheta_{EO} = \left(k_{w,0} \cdot \frac{1}{\overset{*}{m}_{exh}} + k_{w,1} + k_{w,2} \cdot \overset{*}{m}_{exh} \right) \cdot \Delta\vartheta_{w}. \tag{2.46}$$

For the identification of the coefficients $k_{w,i}$, the load step measurements are used since only this type of excitation yields a perceivable dynamic response of the exhaust-gas temperature. Figure 2.21 shows the result. A reduced model with $k_{w,0} = k_{w,2} = 0$ is found to be only slightly less accurate. In this model, the temperature reduction of the gas is proportional to the temperature difference between the gas and the casing.

Transition to a first-order element The stationary heat transfers from the gas to the casing do not have to be considered. The temperature of

the exhaust gas is usually measured at the confluence point. Therefore, the stationary measurements used for the identification of the model for ϑ_{EO} already account for the heat losses occurring between the outlet ports of the engine and the turbine inlet. Similarly, the mass flow and the enthalpy extraction of the turbine are identified using this temperature. For these reasons, it is sufficient to model only the additional heat transfers introduced by transient effects.

From another point of view this simplification means that, during stationary operation, the casing may be assumed to be as hot as the exhaust gas. Therefore, this pseudo casing temperature can be approximated by applying a first-order element to the static exhaust-gas temperature, with the same characteristic time τ_{c} identified for the physical casing.

The original equations (2.41), (2.42) and (2.43) combine to

$$\frac{d\vartheta_{\mathrm{c}}}{dt} = \frac{\bar{\vartheta}_{\mathrm{c}} - \vartheta_{\mathrm{c}}}{\tau_{\mathrm{c}}} = \frac{k_0 + \bar{\vartheta}_{\mathrm{EO}} \cdot \left(k_1 + k_2 \cdot \overset{*}{m}_{\mathrm{exh}} + k_3 \cdot \overset{*}{m}_{exh}^{2} \right) - \vartheta_{\mathrm{c}}}{\tau_{\mathrm{c},0} + \tau_{\mathrm{c},1} \cdot \overset{*}{m}_{\mathrm{exh}}}. \quad (2.47)$$

The introduction of the pseudo casing temperature $\hat{\vartheta}_{\mathrm{c}}$ reduces this expression to

$$\frac{d\hat{\vartheta}_{\mathrm{c}}}{dt} = \frac{\bar{\vartheta}_{\mathrm{EO}} - \hat{\vartheta}_{\mathrm{c}}}{\tau_{\mathrm{c},0} + \tau_{\mathrm{c},1} \cdot \overset{*}{m}_{\mathrm{exh}}}. \quad (2.48)$$

This formulation neglects stationary heat transfers. Accordingly, the simplified model for the reduction of the exhaust-gas temperature becomes

$$\vartheta_{\mathrm{EO}} = \bar{\vartheta}_{\mathrm{EO}} - k_{\mathrm{w},1} \cdot \left(\bar{\vartheta}_{\mathrm{EO}} - \hat{\vartheta}_{\mathrm{c}} \right). \quad (2.49)$$

The stationary exhaust-gas temperature $\bar{\vartheta}_{\mathrm{EO}}$ is calculated by (2.26). For the mass balance in the exhaust manifold in Eq. (2.28c), the average temperature $\vartheta_{\mathrm{EM}} = 0.5 \cdot (\bar{\vartheta}_{\mathrm{EO}} + \vartheta_{\mathrm{EO}})$ is used. This choice corresponds to the assumption that the heat losses are uniformly distributed on the length of the manifold.

Influence on the turbocharger Heat losses also occur throughout the turbine casing. Since the gas temperature could not be measured inside the turbine, the heat transfers to the turbine casing are assumed to exhibit the same dynamics as those in the exhaust manifold. Two options are available to account for these additional losses. The first one is to introduce a factor that reduces the apparent temperature defining the enthalpy available to the turbine,

$$\vartheta_{\mathrm{TB,in}} = \bar{\vartheta}_{\mathrm{EO}} - (1 + k_{\mathrm{TB}}) \cdot k_{\mathrm{w},1} \cdot \left(\bar{\vartheta}_{\mathrm{EO}} - \hat{\vartheta}_{\mathrm{c}}\right). \tag{2.50}$$

A value of $k_{\mathrm{TB}} \approx 0.5$ is found to be physically reasonable. However, an unambiguous identification is difficult since the same effect on the turbocharger dynamics is achieved by increasing $k_{\mathrm{w},1}$ accordingly, which is the second option. Since the effect of this compensation on ϑ_{EO} is small and cannot be discerned in a comparison to measurement data, this option is chosen here.

Validation of the overall model Figure 2.22 displays the results of a simulation of the full air-path model and constrasts them with the measurement data and the model without heat losses. The parameter values identified by the procedures outlined above are used, which read $k_{\mathrm{w},1} = 0.1423$, $\tau_{\mathrm{c},0} = 157.2$, and $\tau_{\mathrm{c},1} = -182.6$

Identification using standard measurements In a usual measurement setup on an engine test-bench, no sensors for the surface temper-

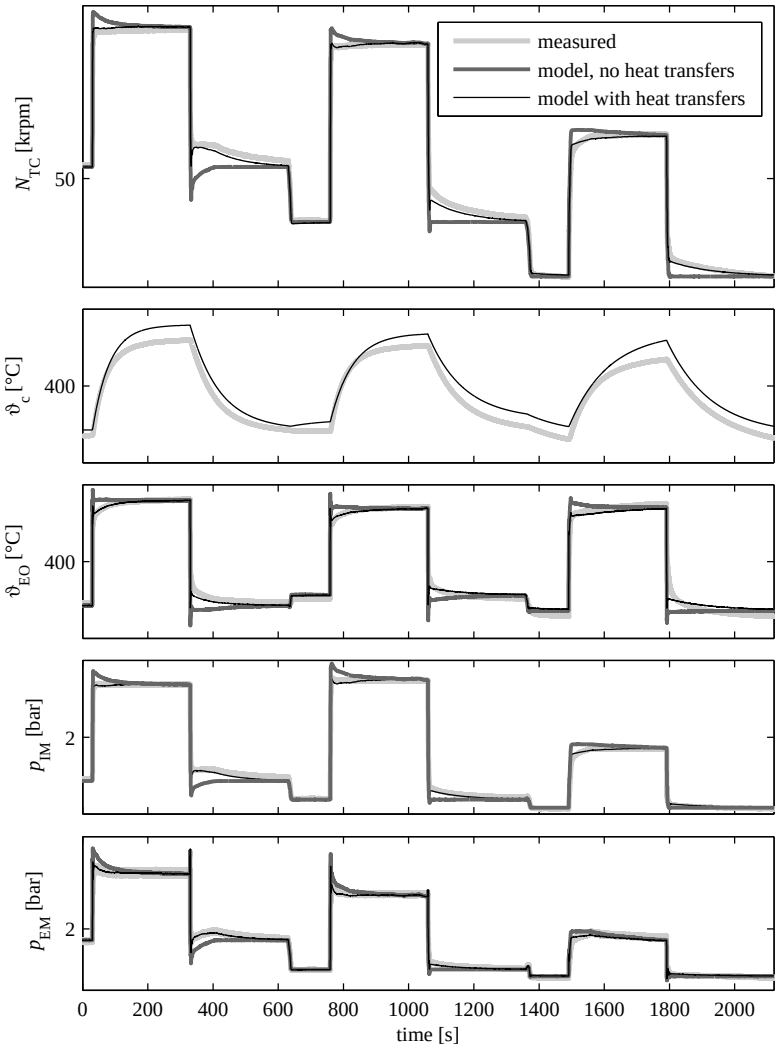

Figure 2.22: Validation of the full air-path model versions without and with the thermal model for the exhaust system. The error in the turbocharger speed at around 500s and 1250s can be attributed to stationary model errors. For the model, the pseudo casing temperature defined by Eq. (2.48) is shown in the second plot. For reasons of confidentiality, the scales are hidden.

ature are available. Moreover, due to the risk of breakage of the fast, but fragile 0.5mm thermocouples, usually slow 3mm thermocouples are used. Therefore, the question left to be answered is whether the model parameters can be identified using only commonly available signals.

One problem that arises is the fact that the stationary errors of the air-path model distort the identification process. One remedy could be to consider the transient segments of the measurement data only, or to band-pass filter the error signal such that the relevant information can be extracted. However, since the thermal dynamics of the exhaust system are very slow with a characteristic time of around 100s, which corresponds to 0.01Hz, this separation is not possible. Another option is to empirically eliminate the stationary error by an appropriate scaling of the signals. However, since the system considered is nonlinear, there is no straightforward way to achieve this correction.

Nevertheless, using the cumulative squared difference between a measured signal and the corresponding model output yields satisfactory results. In this approach, one objective-function evaluation consists in a simulation of the model over the time horizon of the measurement. The turbocharger speed is considered the most significant quantity and is thus used in the objective function. Besides a manual tuning of the parameters, two automated approaches are applicabe.

- *Brute force.* All combinations of the three parameters are calculated on a reasonable grid, and the best combination is selected. A straightforward parallelisation and no reliance on a nonlinear optimisation are the advantages of this approach.

  Figure 2.23 shows the error surface for various values of $\tau_{c,0}$ and $k_{w,1}$ for a constant value of $\tau_{c,1} = -173.5$. The latter is identified as the optimal value by constructing the error surfaces for various values of $\tau_{c,1}$. The lowest error is achieved for $\tau_{c,0} = 176.8$ and

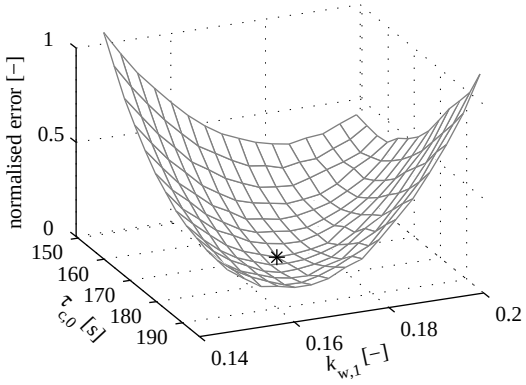

Figure 2.23: Error surface for the brute-force identification of the parameters of the thermal model, $\tau_{c,1} = -173.5$

$k_{w,1} = 0.1672$. These values deviate from those identified using the specific measurements by 12% and 17%, respectively. For $\tau_{c,1}$, the respective figure is 5%.

- *Nonlinear search.* A good initial guess is mandatory when a local nonlinear optimisation is applied to identify the parameters. Therefore, a rough tuning by hand has to be performed beforehand. The optimisation (by `fminsearch` or `fminunc`) always achieves a small value of the objective function. However, the resulting parameters are close to their initial values. This behaviour could be the result of the steady-state errors of the model that distort the objective function.

Since there are few parameters only, it is preferable to adjust them by hand to obtain a physically plausible parameter set that correctly reproduces the qualitative trends of the transient data.

Two further findings are deduced from an application of the model to the engines A and B.[7]

- The loss of model accuracy when a constant τ_c is imposed, i.e. $\tau_{c,1} = 0$, is small. To ease the tuning of the model, the application of a constant thermal capacity is thus preferable.
- The parameter $k_{w,1}$ is nearly invariant for different engines, which is reasonable since the process of heat exchange remains the same. Conversely, the thermal capacity increases with the size of the engine.

Thermal dynamics in the intake manifold Transient measurements indicate that the intake temperature cannot be assumed to always match its stationary value. For the mean-value model itself, the resulting errors of less than 4% have no significant effect on the accuracy of the overall model. However, the NO_x formation is highly sensitive w.r.t. the intake temperature, see Sec. 2.5.5.

The heat transfers in the intake manifold are much smaller than those in the exhaust system. The temperatures and thus the temperature gradients are smaller. However, often a controlled intercooler is used. If the control strategy is not known, the apparent influence on the outlet temperature has to be accounted for. Often, the effect is similar to a first-order element because the controller tries to keep the outlet temperature at a certain reference level.

Figure 2.24 shows the temperature in the intake manifold and the one resulting from an intercooler model which is accurate for stationary operation. When a first-order element

$$\frac{d\vartheta_{IM}}{dt} = \frac{\vartheta_{IC} - \vartheta_{IM}}{\tau_{IC}} \tag{2.51}$$

[7] The results for those engines are presented in Sec. 2.2.4.

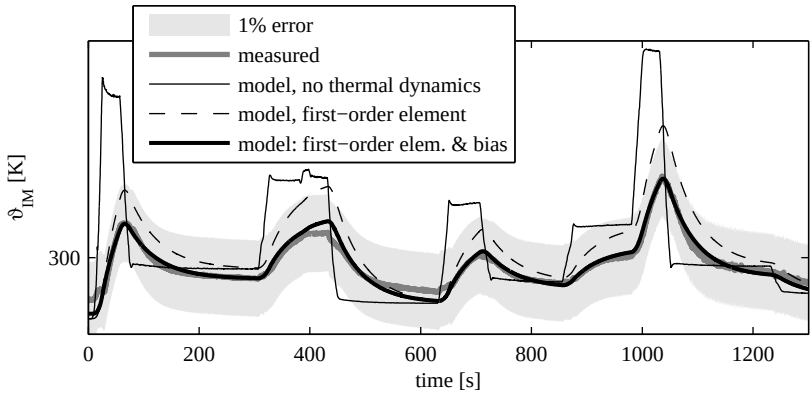

Figure 2.24: Development of the thermal model for the intake manifold. For the model without the thermal dynamics, $\vartheta_{IM} = \vartheta_{IC}$. The scale is hidden for reasons of confidentiality.

with a time constant of $\tau_{IC} = 50$s is applied to this signal, the dynamics are captured accuratey. A bias remains, which may be caused by a different reference temperature set for the intercooler controller.[8] A simple empirical correction may be applied to account for this offset,

$$\vartheta_{IC} = \bar{\vartheta}_{IC} - k_{IC,bias} \cdot (\bar{\vartheta}_{IC} - \vartheta_{cool}). \tag{2.52}$$

The parameters are adjusted by hand, and values of $k_{IC,bias} = 0.3$ and $\vartheta_{cool} = 296$K yield the result shown in Fig. 2.24.

As for the model of the static outlet temperature of the intercooler, its dynamic behaviour also has to be analysed and captured by an appropriate model for each engine individually. The alternative is to include the controller for the intercooler in the air-path model.

[8] The transient measurements were performed on a different day than the stationary ones.

For an engine with EGR, the temperature at the outlet of the EGR cooler has to be considered a dynamic quantity, similar to the temperature at the intercooler outlet. If the dynamics of the two coolers are similar, the two temperatures can be linked in order to avoid adding an additional state variable. The temperature of the mixture of fresh air and recirculated exhaust gas is

$$\bar{\vartheta}_{\mathrm{IM,in}} = \frac{\vartheta_{\mathrm{IC}} \cdot \overset{*}{m}_{\mathrm{CP}} + \vartheta_{EGR} \cdot \overset{*}{m}_{\mathrm{EGR}}}{\overset{*}{m}_{\mathrm{CP}} + \overset{*}{m}_{\mathrm{EGR}}}. \tag{2.53}$$

The dynamic compensation (2.51) is applied to this average temperature,

$$\frac{d\vartheta_{\mathrm{IM,in}}}{dt} = \frac{\bar{\vartheta}_{\mathrm{IM,in}} - \vartheta_{\mathrm{IM,in}}}{\tau_{\mathrm{IC}}}, \tag{2.54}$$

and the balance equations for the mass (2.31a) and the energy (2.31b) become

$$\frac{dp_{\mathrm{IM}}}{dt} = \frac{R \cdot \kappa}{V_{\mathrm{IM}}} \cdot \left((\overset{*}{m}_{\mathrm{CP}} + \overset{*}{m}_{\mathrm{EGR}}) \cdot \vartheta_{\mathrm{IM,in}} - \overset{*}{m}_{\mathrm{cyl}} \cdot \vartheta_{\mathrm{IM}} \right) \tag{2.55a}$$

$$\frac{d\vartheta_{\mathrm{IM}}}{dt} = \frac{R \cdot \vartheta_{\mathrm{IM}}}{p_{\mathrm{IM}} \cdot V_{\mathrm{IM}} \cdot c_{\mathrm{v}}} \cdot \Big[c_{\mathrm{p}} \cdot \left((\overset{*}{m}_{\mathrm{CP}} + \overset{*}{m}_{\mathrm{EGR}}) \cdot \vartheta_{\mathrm{IM,in}} - \overset{*}{m}_{\mathrm{cyl}} \cdot \vartheta_{\mathrm{IM}} \right) - c_{\mathrm{v}} \cdot \vartheta_{\mathrm{IM}} \cdot \left(\overset{*}{m}_{\mathrm{CP}} + \overset{*}{m}_{\mathrm{EGR}} - \overset{*}{m}_{\mathrm{cyl}} \right) \Big]. \tag{2.55b}$$

2.2.4 Transient validation

The full air-path models are validated using measurements on the World Harmonized Transient Cycle (WHTC). The cycle is presented and analysed in more detail in Sec. 4.3. The engine speed and load torque for the segment shown here are depicted in the right-hand plot of Fig. 2.25. The left-hand plot shows the first two operating-point changes of the

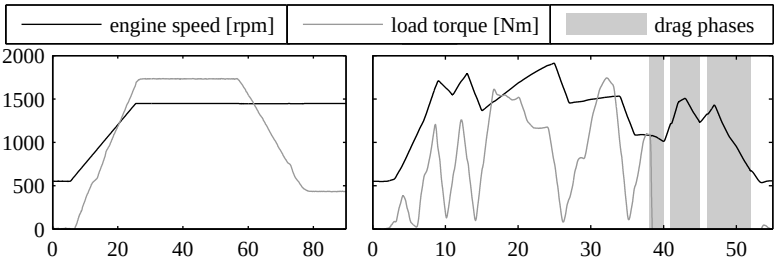

Figure 2.25: Cycles used for the transient validation of the air-path model, scaled for engine A.

World Harmonized Stationary Cycle (WHSC). This profile is used to illustrate the effect of the thermal models more clearly.

Figure 2.26 shows the results of the model for engine A. The model structure without the additional restrictions for the AFM and the ATS is used. The model without the thermal capacities in the intake and the exhaust manifolds is able to accurately reproduce the fast dynamics defined by the turbocharger inertia and the volumes. However, the turbocharger speed, the pressures, and the intake temperature are overestimated whenever the load increases. The inclusion of the thermal models substantially improves the model w.r.t. this effect.

The results for engine B are shown in Fig. 2.27. Only the segment of the WHTC is shown. The engine with EGR is considered. The results for the engine without EGR are similar to those of engine A. Also, the inclusion of the AFM and ATS restrictions in the case of no EGR does not have any impact on the model quality. Therefore, this extension is beneficial only if the model has to be able to predict the influence of varying ambient conditions such as the low pressure faced at a high altitude.

The temperature in the intake manifold is not predicted as accurately as for engine A. Since the controller of the EGR cooler also adapts

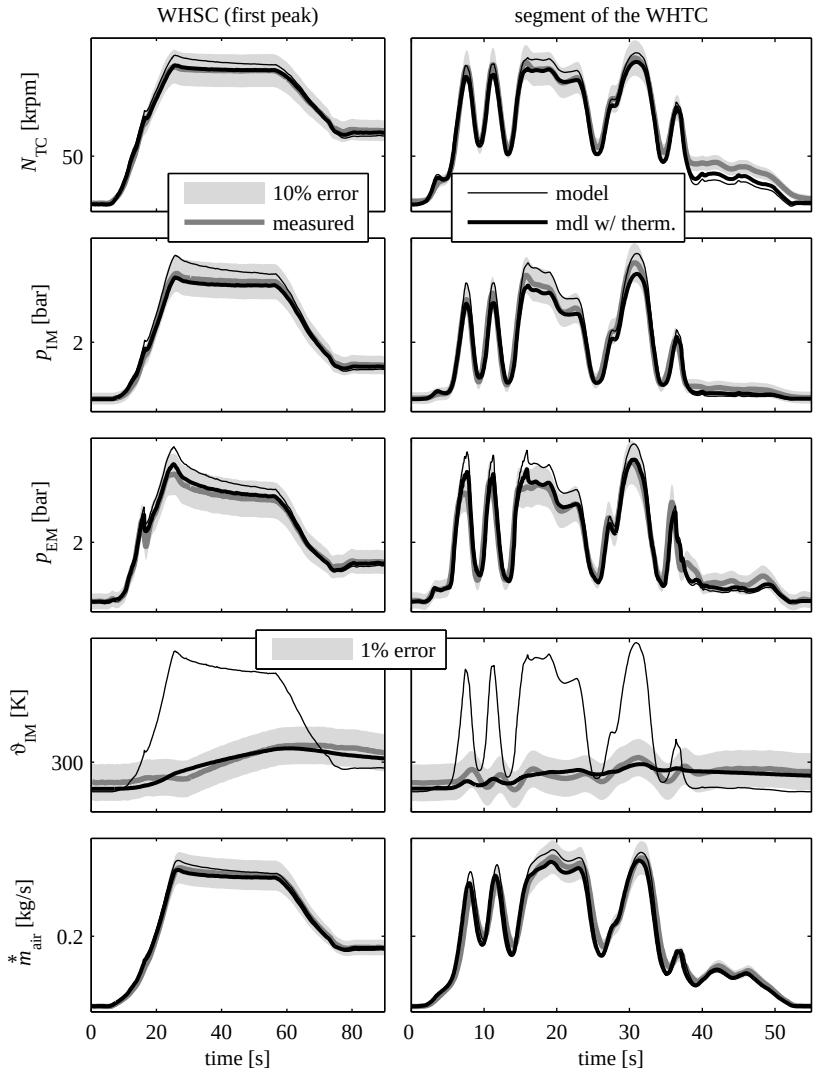

Figure 2.26: Validation of the air-path model for engine A, model structure without AFM and ATS. For the model, the compressor mass-flow is shown in the last plot since the mass flow through the AFM is not modelled. For reasons of confidentiality, the scales are hidden.

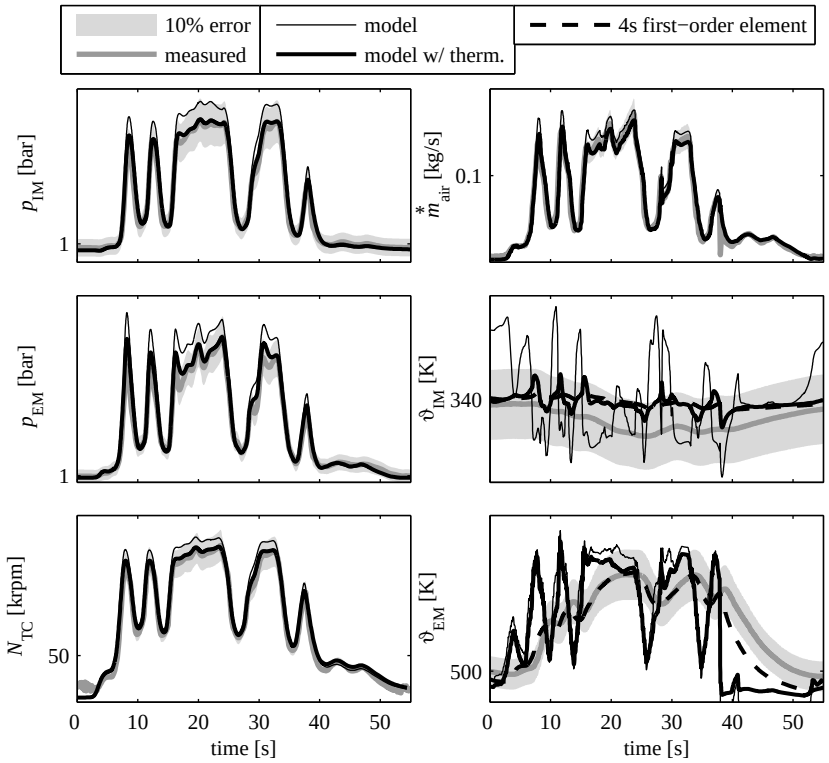

Figure 2.27: Validation of the air-path model for engine B, with EGR. For reasons of confidentiality, the scales are hidden.

its setpoint, a time-variable offset to the model identified by stationary measurements is introduced. An assessment of the model for the exhaust-gas temperature is not possible using the data obtained by the standard, slow thermocouples. As indicated in the corresponding plot, these sensors exhibit very slow dynamics that can be approximated by a first-order element with a time constant of 4 s.

All models are able to accurately reproduce the engine behaviour even for idle conditions or during drag phases. Therefore, they are applicable for the simulation and the optimal control over full driving cycles, including idling and drag phases.

2.3 Torque generation

Two formulations of the model for combustion efficiency are possible. These two cases are related to the fact that the engine speed is either prescribed, e.g. when the engine is operated on a test bench, or calculated dynamically, e.g. when the model is used in a simulation of an entire drivetrain or vehicle. The former is termed "inverted" formulation since the fuel mass is calculated from the prescribed load torque and engine speed. In contrast, the "causal" formulation considers the fuel mass as an additional input to the model. The inverted formulation is presented here, but any divergence of the causal version from its inverted counterpart is indicated in the text.

The approach used here is adapted from [92]. A base efficiency is corrected by multiplicative terms which account for the reduction of the combustion efficiency due to a deviation of the relevant inputs from their efficiency-optimal values. The model considers the influences of the boost level, represented by the density of the intake air [102], the injection timing, represented by the start of the main injection SOI, and the common-rail pressure.

$$\eta = \eta_0(N_{\text{eng}}, m_{\text{fcc}}) \cdot \eta_\rho \cdot \eta_{\text{SOI}} \cdot \eta_{\text{rail}}. \tag{2.56}$$

By adding further multiplicative efficiency-reduction factors, the model can be extended to account for further influences such as EGR, see Sec. 2.3.1 below, or an injection splitting.

Engine torque To produce the prescribed load torque, the engine torque[9] has to compensate for several losses as well as take into account

[9] Alternatively, the engine torque is often termed the inner torque.

the acceleration of the engine,

$$T_{\mathrm{eng}} = T_{\mathrm{load}} + T_{\mathrm{fric}} + T_{\mathrm{gex}} + T_{\mathrm{HPP}} + \Theta_{\mathrm{eng}} \cdot \frac{\pi}{30} \cdot \frac{dN_{\mathrm{eng}}}{dt}. \tag{2.57}$$

The friction torque is modelled by a quadratic polynomial in the engine speed,

$$T_{\mathrm{fric}} = T_{\mathrm{f},0} + T_{\mathrm{f},1} \cdot N_{\mathrm{eng}} + T_{\mathrm{f},2} \cdot N_{\mathrm{eng}}^2. \tag{2.58}$$

The gas-exchange losses are imposed by the pressure difference between the intake and exhaust manifolds,

$$T_{\mathrm{gex}} = (p_{\mathrm{EM}} - p_{\mathrm{IM}}) \cdot \frac{V_{\mathrm{d}}}{4\pi}. \tag{2.59}$$

Since the high-pressure pump (HPP) of the common-rail system can drain up to 2% of the engine power, this auxiliary has to be accounted for as well. The power required to drive a mass flow $\overset{*}{m}$ at pressure p is

$$P = F \cdot v = p \cdot A \cdot v = p \cdot \frac{\overset{*}{m}}{\rho}. \tag{2.60}$$

Inserting the engine-related quantities and rearranging the terms yields

$$T_{\mathrm{HPP}} = \frac{p_{\mathrm{rail}} \cdot \tilde{m}_{\mathrm{fcc}}}{\eta_{\mathrm{HPP}} \cdot \rho_{\mathrm{fuel}}} \cdot \frac{n_{\mathrm{cyl}}}{4\pi}, \tag{2.61}$$

where $\eta_{\mathrm{HPP}} = 0.75$ is the estimated pump efficiency and $\rho_{\mathrm{fuel}} \approx 840^{\mathrm{kg/m^3}}$ is the density of diesel fuel, which is assumed to be incompressible. The fuel mass, which is not known as yet, is approximated by an affine function of the desired load torque,

$$\tilde{m}_{\mathrm{fcc}} = \tilde{m}_{\mathrm{f},0} + \tilde{m}_{\mathrm{f},1} \cdot T_{\mathrm{load}}. \tag{2.62}$$

The error of a few percentage points is negligible considering the ratio of the HPP torque to the load torque. In the case of the causal formulation, this approximation is not required.

The thermal efficiency is defined as the quotient of the mean effective pressure p_{me} and the energy introduced by the fuel, expressed as the "fuel mean pressure" p_{mf}.

$$\eta := \frac{p_{\mathrm{me}}}{p_{\mathrm{mf}}} = \frac{T_{\mathrm{eng}} \cdot 4\pi}{m_{\mathrm{fcc}} \cdot n_{\mathrm{cyl}} \cdot H_{\mathrm{l}}} = \frac{Q_{\mathrm{work}}}{n_{\mathrm{cyl}} \cdot Q_{\mathrm{fuel}}}. \tag{2.63}$$

It relates the engine torque to the fuel mass and thus appears in both the inverted and the causal formulation.

Multiplicative efficiency reductions All factors in Eq. (2.56) are of the form

$$\eta_i = 1 - a_i \cdot \| u_i - u_{i,\mathrm{opt}}(N_{\mathrm{eng}}, T_{\mathrm{eng}}) \|^{b_i}. \tag{2.64}$$

The reference inputs $u_{i,\mathrm{opt}}$ are the optimal values for the inputs u_i w.r.t. efficiency and can be represented by polynomials, along with the corresponding peak efficiency η_0. The notation is simplified using $N := N_{\mathrm{eng}}$ and $T := T_{\mathrm{eng}}$. In the causal formulation, T_{eng} is replaced by m_{fcc} in Eqs. (2.64)-(2.65d).

$$\rho_{\mathrm{opt}} = \rho_0 + \rho_1 \cdot N + \rho_2 \cdot T + \rho_3 \cdot N \cdot T, \tag{2.65a}$$

$$\varphi_{\mathrm{SOI,opt}} = \varphi_0 + \varphi_1 \cdot N + \varphi_2 \cdot T + \varphi_3 \cdot N \cdot T, \tag{2.65b}$$

$$p_{\mathrm{rail,opt}} = p_{\mathrm{r},0} + p_{\mathrm{r},1} \cdot N + p_{\mathrm{r},2} \cdot T + p_{\mathrm{r},3} \cdot N^2, \tag{2.65c}$$

$$\eta_0 = e_0 + e_1 \cdot N + e_2 \cdot T + e_3 \cdot T^2 + e_4 \cdot N \cdot T. \tag{2.65d}$$

The identification procedure is similar to the one applied for the ignition-delay model, see Sec. 2.4.3 below. Figure 2.28 shows the mea-

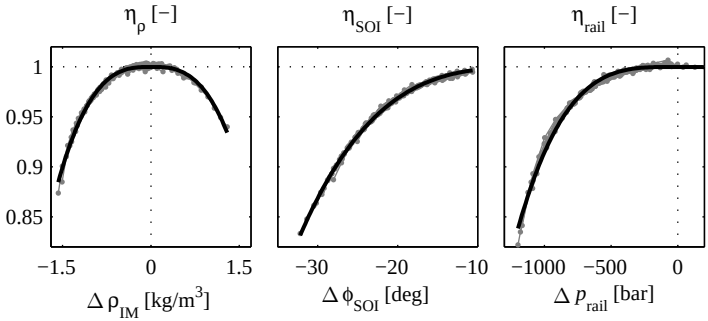

Figure 2.28: Individual multiplicative efficiency reductions, engine A. Measured (grey) in all 16 variation operating points, and model (black).

sured efficiency-reduction factors and the corresponding model fits for engine A. The model is able to represent all data with a standard deviation of 0.6% and a maximum relative error magnitude of 2%. The cross-variations are not used for the identification. For the other engines, a similar model accuracy is achieved. For the causal formulation and for different engines, slightly different regressors need to be chosen for the base maps in Eq. (2.65).

2.3.1 Extension to EGR

The reduction of the combustion efficiency due to EGR is only minor. At commonly used EGR rates, a reduction of around 3% is observed for engine B. Only if the EGR valve is completely opened and the exhaust flap is fully closed, a reduction of around 7% results in some specific operating points. Compared to the reduction by up to 15% effected by the other control inputs, these figures are small. Nevertheless, to account for the tradeoff between fuel consumption and NO_x emissions in the OCP, this effect has to be included in the model.

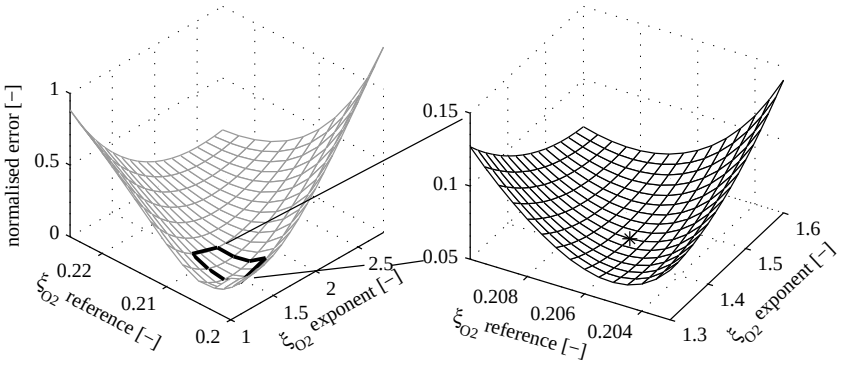

Figure 2.29: Error surfaces for the identification of the two parameters defining the efficiency reduction due to limited oxygen availability (EGR). The optimal values obtained are 0.206 for the reference and 1.42 for the exponent.

A simple extension of the model presented above is applied. Similarly to the other multiplicative reduction factors, the factor

$$\eta_{\xi_{O_2}} = 1 - a_{\xi_{O_2}}(N_{\text{eng}}, m_{\text{fcc}}) \cdot \|\xi_{O_2} - \hat{\xi}_{O_2}\|^{b_{\xi_{O_2}}}. \tag{2.66}$$

is appended to Eq. (2.56). The exponential coefficient $b_{\xi_{O_2}}$ and the reference oxygen mass-fraction $\hat{\xi}_{O_2}$ are identified by a nonlinear search. In the objective function, the coefficient $a_{\xi_{O_2}}$ is chosen individually for each operating point by an LSQ regression. The error surfaces of the nonlinear objective function are shown in Fig. 2.29.

The reduction coefficient is approximated by

$$a_{\xi_{O_2}} = a_0 + a_1 \cdot N_{\text{eng}} + a_2 \cdot m_{\text{fcc}}. \tag{2.67}$$

The three parameters a_i are first identified by an LSQ regression using the operating-point individual values. Subsequently, along with the two

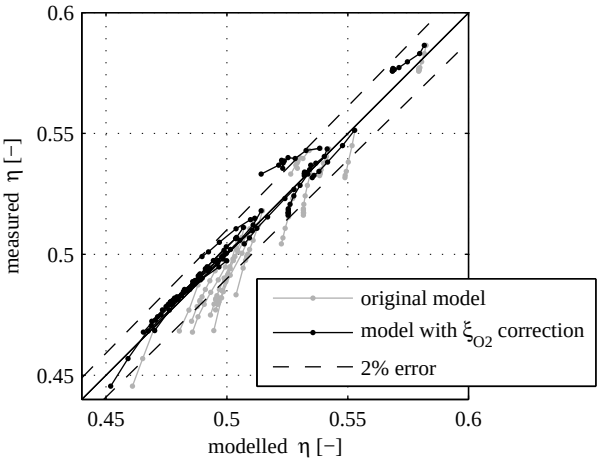

Figure 2.30: Results of the extension of the efficiency model towards EGR. With the exception of one operating point, the trend of the effect is accurately captured by the model.

nonlinear parameters, they are fine-tuned by a nonlinear optimisation. Figure 2.30 displays the results of the model.

2.4 Cylinder model

In this section, the submodels are derived that serve as an interface between the air-path quantities and the NO_x-formation model. A semi-empirical approach is used, which means that the known qualitative trends of the underlying physical processes are approximated directly. The quantitative accuracy is achieved by fitting the model to a specific set of measurement data. This modelling approach is well suited for phenomena which consist of multiple complex and interrelated processes but exhibit a comprehensible high-level behaviour.

Note that if an engine contains specific additional features such as a swirl valve or a variable valve-train, the models presented here have to be extended to include their effects. The purpose of this section is to demonstrate the possibility of deriving simple, yet sufficiently accurate models and to provide a starting point for possible extensions. Models are needed for the composition of the cylinder charge, the compression, and the ignition delay.

2.4.1 Composition of the cylinder charge

The flow of the mixture in the intake manifold into the cylinders, $\overset{*}{m}_{\mathrm{cyl}}$, is calculated by Eq. (2.21). From this mass flow, the mass of fresh mixture in each cylinder is calculated as

$$m_{\mathrm{in}} = \overset{*}{m}_{\mathrm{cyl}} \cdot \frac{60}{N_{\mathrm{eng}}} \cdot \frac{2}{n_{\mathrm{cyl}}}. \tag{2.68}$$

Two more quantities are required to define the composition and the state of the cylinder charge when the intake valve closes (IVC): The pressure at IVC and the mass of the residual gas. The latter is the burnt gas that remains in the cylinder after the combustion.

The pressure in the cylinder at IVC is modelled as an affine function of the pressure in the intake manifold. Furthermore, a slight depen-

dency on the engine speed can be observed,

$$p_{\mathrm{IVC}} = k_{\mathrm{p},0} + p_{\mathrm{IM}} \cdot (k_{\mathrm{p},1} + k_{\mathrm{p},2} \cdot N_{\mathrm{eng}}). \tag{2.69}$$

A standard deviation of below 0.7% results for engines A and B.

The mass of exhaust gas remaining in the cylinder is defined mainly by the density of the exhaust gas in the exhaust manifold. It defines the gas mass that remains in the combustion-chamber volume, which is not swept by the piston. However, due to overlapping valve timings, the pressure difference across the engine and the engine speed affect the mass of residual gas as well. Thus, the regressors

$$m_{\mathrm{RG}} = k_0 + k_1 \cdot \rho_{\mathrm{EM}} + k_2 \cdot (p_{\mathrm{EM}} - p_{\mathrm{IM}}) + k_3 \cdot N_{\mathrm{eng}} + k_4 \cdot N_{\mathrm{eng}}^2 \tag{2.70}$$

are used as a model, where $\rho_{\mathrm{EM}} = p_{\mathrm{EM}}/(287 \cdot \vartheta_{\mathrm{EM}})$. The variations as well as the nominal map can be represented with a relative standard deviation of 3.8% for engine A. For engines in which valve overlap does not play a significant role, such as engine B, the first two regressors are sufficient as a model.

The total mass in the cylinder is $m_{\mathrm{cyl}} = m_{\mathrm{in}} + m_{\mathrm{RG}}$, and the residual-gas fraction is defined as $x_{\mathrm{RG}} = m_{\mathrm{RG}}/m_{\mathrm{cyl}}$. Based on these quantities and on the composition of the fresh mixture in the intake manifold, ξ_{in}, the gas compositions before and after combustion, ξ_{bC} and ξ_{aC}, are calculated as described in Appendix B.

2.4.2 Compression

The pressure and the temperature at start of combustion (SOC) are inputs to the NO_x-formation model. The crank angle at which combustion begins is predicted by the ignition-delay model described in the

next section. The pressure is estimated by the polytropic relation

$$p_{\mathrm{SOC}} = p_{\mathrm{IVC}} \cdot \left(\frac{V_{\mathrm{IVC}}}{V_{\mathrm{SOC}}} \right)^{\kappa_{\mathrm{c}}}. \tag{2.71}$$

For the relevant temperature range of 300-950 K, the temperature dependency of the polytropic exponent κ_{c} can be closely approximated by an affine function. By application of the ideal-gas law, the temperature is represented by pV/mR_{c}. The gas constant R_{c} of the cylinder charge, which is defined by the gas composition, varies between 283.65 and 284.05 for all measurement data available. Thus, by using its average value, an error of less than 0.07% is introduced. With this simplification, the only variable remaining in the expression for κ_{c} which does not occur in the polytropic relation anyway is the mass. Therefore, the approximation

$$\kappa_{\mathrm{c}} = \kappa_{\mathrm{c},0} + \kappa_{\mathrm{c},1} \cdot \frac{1}{m_{\mathrm{cyl}}} \tag{2.72}$$

is used. The temperature at SOC is calculated from the corresponding pressure by the ideal-gas law

$$\vartheta_{\mathrm{SOC}} = p_{\mathrm{SOC}} \cdot V_{\mathrm{SOC}} / (m_{\mathrm{cyl}} \cdot R_{\mathrm{c}}). \tag{2.73}$$

For each measurement, the “perfect” coefficient can be calculated by solving Eq. (2.71) for κ_{c}. The resulting coefficients are used to identify the two parameters in Eq. (2.72) by a linear least-squares regression. For engine A, the resulting relative error in the pressure and temperature at SOC exhibits a standard deviation of 0.65%. For comparison, a constant polytropic coefficient, chosen as the average of the “perfect” coefficients, leads to systematic errors of up to 6.2%, with an offset of +0.59% and a standard deviation of 1.51%.

2.4.3 Ignition delay

The most common way of modelling the ignition delay is by integrating the formation rate of radicals, which lead to ignition when their concentration exceeds a critical level. The formation rate is approximated by an Arrhenius function of the form

$$r_{\mathrm{ID}} = \frac{1}{t_{\mathrm{ID}}} = A \cdot p^b \cdot \exp\left(-\frac{\vartheta_{\mathrm{A}}}{\vartheta}\right). \tag{2.74}$$

To obtain the crank angle φ_{SOC} at which ignition occurs, this formation rate has to be integrated until a critical value is reached, which is normalised to unity by the constant A,

$$\frac{1}{6 \cdot N_{\mathrm{eng}}} \cdot \int_{\varphi_{\mathrm{SOI}}}^{\varphi_{\mathrm{SOC}}} \frac{1}{t_{\mathrm{ID}}(\varphi)}\, d\varphi = 1. \tag{2.75}$$

More details on the ignition delay and more complex models are provided in [102, 23]. The goal is to avoid this implicit formulation and its expensive calculation. Moreover, the measurement data at hand could not be reproduced accurately by such an approach.

An operating-point dependent reference value and multiplicative corrections for each input considered are used instead.

$$t_{\mathrm{ID}} = \frac{\varphi_{\mathrm{SOC}} - \varphi_{\mathrm{SOI}}}{6 \cdot N_{\mathrm{eng}}} = t_{\mathrm{ID},0}(N_{\mathrm{eng}}, m_{\mathrm{fcc}}) \cdot t_{\mathrm{p}} \cdot t_{\mathrm{SOI}} \cdot t_{\mathrm{rail}}. \tag{2.76}$$

The influences of boost level, injection timing and injection pressure are interdependent. Figure 2.31 shows an illustrative example. Consider the case where injection starts before top dead center and ignition occurs right at top dead center (t_{1a}). In this case, postponing the injection reduces the ignition delay since the average pressure is higher due to compression (t_{1b}). By reducing the boost pressure (or injection pressure), the ignition delay is prolonged such that ignition occurs later

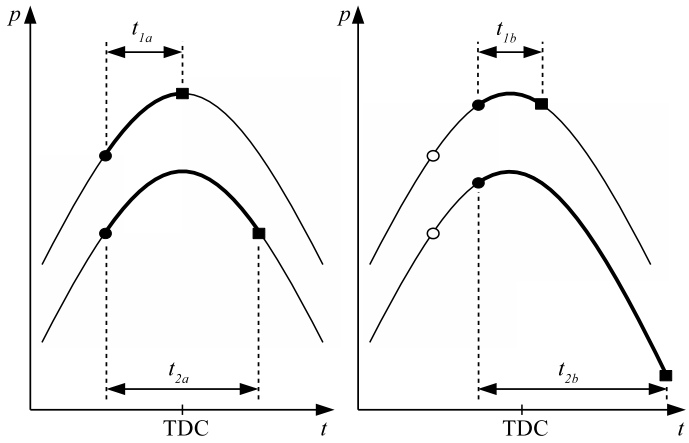

Figure 2.31: Influence of retarding the injection at different pressure levels: the ignition delay is reduced at the high pressure level ($t_{1a} \rightarrow t_{1b}$) but is increased at the low pressure level ($t_{2a} \rightarrow t_{2b}$) when the injection is retarded.

(t_{2a}). Retarding the injection in this case even increases the ignition delay (t_{2b}). The shift towards late combustion replaces the initial high-pressure phase by an extended end phase at lower pressure. Of course, this behaviour changes when the initial SOI is altered.

These considerations challenge the assumption of multiplicative influences and the corresponding model structure (2.76). However, the error introduced by neglecting the cross-influences can be reduced. For instance, largely decoupled inputs can be selected and, when the corresponding setpoints and correction functions are defined, the underlying physical processes have to be kept in mind. In the next paragraphs, these principles are applied to the influences considered here.

Boost Due to the intercooler, the temperature in the intake manifold may be assumed to be approximately constant. Furthermore, the

residual-gas fraction is small and does not vary significantly. Thus, the temperature at IVC is almost constant and the pressure at IVC is chosen to represent boost. The alternative of using the pressure at SOI would not represent the boost level only, but would be a combination of boost level and injection timing.

The ignition delay becomes long at low pressure levels and it is assumed to approach a minimal value for increasingly high pressures. An asymptotic correction function is thus proposed,

$$t_{\mathrm{p}} = t_{\mathrm{p},0} + \exp\left(-t_{\mathrm{p,e}} \cdot p_{\mathrm{IVC}}\right). \tag{2.77}$$

Rail pressure Similar considerations hold for the injection pressure. Due to the turbulence introduced, spray break-up, fuel evaporation and also mixing are faster at a higher injection pressure. The model structure chosen is thus identical to the one selected for the boost level.

$$t_{\mathrm{rail}} = t_{\mathrm{rail},0} + \exp\left(-t_{\mathrm{rail,e}} \cdot p_{\mathrm{rail}}\right). \tag{2.78}$$

Injection timing The cylinder volume around top dead-center can be represented by a quadratic function. Therefore, expressing the pressure and the temperature by the polytropic relation enables the approximation of the integrand in (2.75) by a function which basically is a modified quadratic function. Its primitive thus resembles a cubic polynomial. Consequently, the correction function for the injection timing is chosen as a cubic polynomial in the deviation from an optimal timing which depends on the operating point. Since the linear term has been found to be negligible, the final equation reads

$$\begin{aligned} t_{\mathrm{SOI}} &= 1 + t_{\mathrm{SOI},2} \cdot \Delta\varphi^2 + t_{\mathrm{SOI},3} \cdot \Delta\varphi^3, \\ \Delta\varphi &= \varphi_{\mathrm{SOI}} - \varphi_{\mathrm{opt}}(N_{\mathrm{eng}}, m_{\mathrm{fcc}}). \end{aligned} \tag{2.79}$$

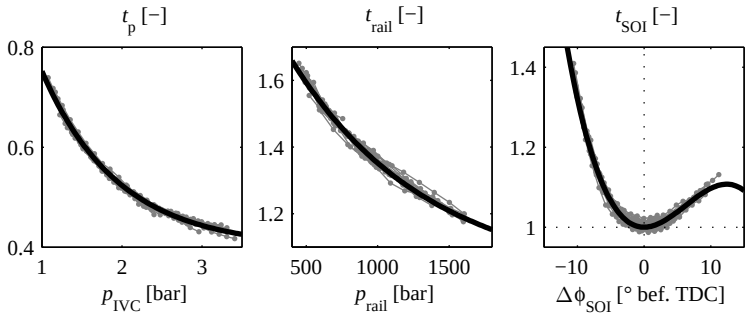

Figure 2.32: Individual multiplicative correction factors of the ignition-delay model, measured (grey) in all 16 variation operating points and model (black).

The two maps, namely the reference ignition delay $t_{ID,0}$ and the optimal SOI φ_{opt}, can be represented by linear functions in the engine speed and fuel fuel mass,

$$t_{ID,0} = k_{t,0} + k_{t,1} \cdot N_{eng} + k_{t,2} \cdot m_{fcc}, \tag{2.80}$$

$$\varphi_{opt} = k_{\varphi,0} + k_{\varphi,1} \cdot N_{eng} + k_{\varphi,2} \cdot m_{fcc} + k_{\varphi,3} \cdot N_{eng} \cdot m_{fcc}. \tag{2.81}$$

During the identification of the parameters $t_{x,i}$, the two reference quantities are chosen operating-point individually. The measured variations in all 16 variation points and the model fits are shown in Fig. 2.32. Subsequently, the approximations (2.80) and (2.81) are fitted to the resulting values at the variation points and then fine-tuned along with all other parameters. The initial data from the variation-point individual identification and the final polynomial maps are shown in Fig. 2.33. All data, including the cross-variations not used for identification, is represented with a standard deviation of 1.3%, and the largest relative error has a magnitude of 4.1%.

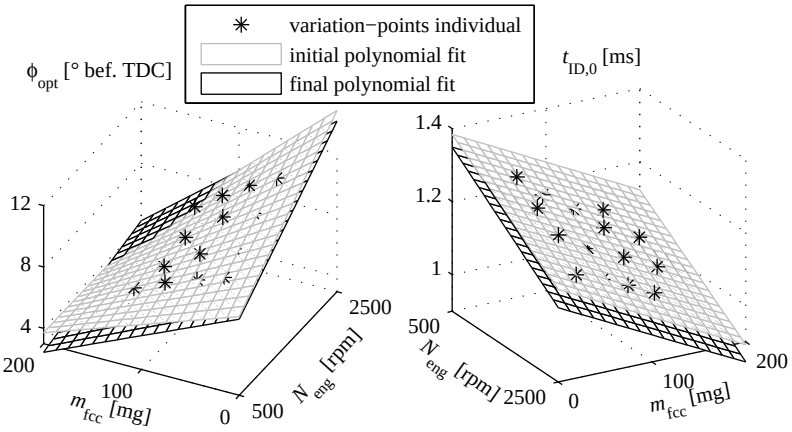

Figure 2.33: Reference values identified individually for each variation point, the polynomial fitted to this data, and the final polynomial maps.

2.4.4 Extension to EGR

The compression model introduced in Sec. 2.4.2 is found not to yield acceptable results for engine B and the version of engine A used for the experimental validation. An alternative model formulation approximates the polytropic coefficient as a function of the engine speed and the pressure at IVC,

$$\kappa_c = \tilde{\kappa}_{c,0} + \tilde{\kappa}_{c,1} \cdot N_{eng} + \tilde{\kappa}_{c,2} \cdot N_{eng}^2 + \tilde{\kappa}_{c,3} \cdot p_{IVC} + \tilde{\kappa}_{c,4} \cdot p_{IVC}^2 + \tilde{\kappa}_{c,5} \cdot N_{eng} \cdot p_{IVC}. \tag{2.82}$$

The model surface identified for engine A used for the experimental validation is shown in Fig. 2.34.

The ignition-delay model is extended by another correction term that accounts for the oxygen availability. The additional correction

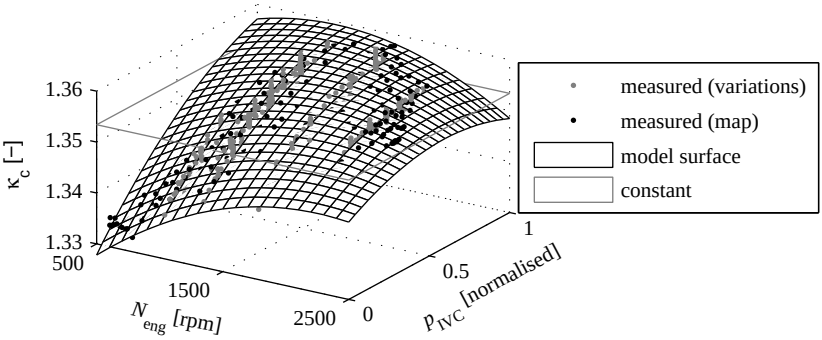

Figure 2.34: Model for the polytropic coefficient for compression of the cylinder charge from IVC to SOC. The data from the configuration of engine A used for the experimental validation is shown.

factor reads

$$t_{\xi_{O_2}} = 1 - k_{\text{lin}} \cdot (\xi_{O_2} - \hat{\xi}_{O_2}) - k_{\text{quad}}(N_{\text{eng}}, m_{\text{fcc}}) \cdot (\xi_{O_2} - \hat{\xi}_{O_2})^2, \quad (2.83a)$$

where the coefficient of the second-order term is a function of the engine operating-point,

$$\begin{aligned} k_{\text{quad}} = {} & k_{\text{quad},0} + k_{\text{quad},1} \cdot N_{\text{eng}} + k_{\text{quad},2} \cdot m_{\text{fcc}} + \\ & k_{\text{quad},3} \cdot N_{\text{eng}}^2 + k_{\text{quad},4} \cdot m_{\text{fcc}}^2 + k_{\text{quad},5} \cdot N_{\text{eng}} \cdot m_{\text{fcc}}. \end{aligned} \quad (2.83b)$$

All data, including the EGR sweeps and all cross variations, is reproduced with a standard deviation of 3.48% by this extended model.

2.5 NO_x emissions

The development of an optimisation-oriented model for the NO_x emissions of diesel engines is one of the principal contributions of this thesis. The presentation of the model in this section closely follows [18] and [19]. First, a literature survey presents the state of the art and related work. Subsequently, the general modelling approach is outlined, before the actual model is described. Finally, the identification procedure is described and a sensitivity analysis performed in order to embedd the model in an air-path model. A more detailed analysis of the model when applied to engine B with EGR, and a statement of the limitations of the model concludes this section.

2.5.1 Literature survey

Models for the nitrogen-oxide (NO_x) emissions of diesel engines may be split into two main groups, namely phenomenological and empirical models. Notice that the class of spatially resolved models [110, 169] used to investigate turbulent-flame phenomena and to perform pre-experimental studies is excluded from this review due to prohibitively high computational costs.

The term “phenomenological” already indicates the inherent difficulty of describing the emission formation using physical first principles only. In fact, the complex nature of diesel combustion and pollutant formation described in [98, 135, 59, 60] requires macroscopic phenomena to be modelled directly. First promising attempts date back to the early ’70s [116, 167]. Together with later contributions [104, 120, 13, 27, 28], they provided a generally accepted framework comprising the following key components: Subdivision of the injection spray into individual fuel packages, calculation of spray breakup, ignition and combustion, and superposition of the NO-formation kinetics using the well-known (extended) Zeldovich mechanism. The tracking of multiple packages

through these phases motivated the term quasidimensional to designate this type of model. More recent research has focused on two main aspects: On one hand, the required detail of the formation mechanism is examined. More complex [106, 170, 122, 100, 101] as well as simple single-equation approaches have been investigated [65, 172, 14, 15, 189]. On the other hand, the main phenomena defining the evolution of temperature and composition of the packages over crank angle are elaborated. The initial temperature is commonly assumed to be the adiabatic flame-temperature (AFT). Its calculation is sometimes simplified by considering the main influence of oxygen availability only [97, 67, 68] or by treating dissociation effects as additional fitting parameters of the model [14, 15, 12]. Polytropic compression and expansion is certainly the most influential factor defining the further course of temperature. More controversial is the mixing of combustion products with unburned gas. Often, mixing is modelled empirically and fitted to match measurement results [67, 68, 182, 187] or is assumed to be driven by the difference between local and global air-to-fuel ratios [122, 180, 108] or by turbulence [121].

Purely empirical models, on the other hand, merely rely on measured data to identify the relevant relations. Some insight into the broad field of empirical emission modelling is given by [105, 164]. Due to their ability of approximating arbitrary functions, neural networks are often chosen as the underlying model structure [118, 93, 25, 24]. Evolutionary (or genetic) algorithms are another promising approach to extract relations from data and are often paired with automatic input-selection algorithms [61, 34]. Hybrid approaches attempt to combine physical considerations and empirical elements. To this end, empirical models are improved by using quantities derived from measured cylinder pressure as inputs [173, 77, 185], or parts of phenomenological models are empirically fitted to match measurement data [14, 15, 70]. Even using

the output of a physical model as input to an empirical fit has been proposed [172, 12, 154].

Phenomenological and empirical approaches both feature desirable properties but also exhibit specific inherent drawbacks. It is an often stated fact that phenomenological models are pertinent for predicting qualitative trends but suffer from error propagation [184, 162] and high sensitivities between sequential submodels [14, 15, 133]. The combination of these properties results in a poor quantitative accuracy with an average magnitude of the relative error of around 15% [167, 14, 15, 187, 121, 90, 8]. Moreover, phenomenological models are difficult to identify [182, 8] and achieve computational speeds of approximately real-time only [108, 70, 133, 7]. In contrast, empirical models are fast, match measurement results accurately and are simple to handle. Their major drawbacks are the lack of reliable far-field extrapolation and the inherent property that only those influences that are explicitly excited during the measurement process can be identified. These restrictions limit the validity of the model to the proximity of the underlying engine calibration and necessitate a fine sampling of this region.

2.5.2 Requirements, procedure and assumptions

The goal of this work is to introduce a novel NO_x-emission model which is particularly well suited for use in computationally intensive fields such as numerical optimal control and model-based engine-calibration and control-design. The model has to meet all requirements stated in the introduction, which essentially represent the combination of the desirable properties of phenomenological and empirical models summarised in the following list.

1. Quantitative accuracy, also during transients;
2. Representation of all qualitative trends throughout the entire input space;
3. Consistent and smooth interpolation;
4. Physically plausible extrapolation;
5. Low measurement burden and usage of standard measurement equipment only to obtain the required data;
6. Fully automated and reproducible identification of the model parameters;
7. Fast execution speed, i.e. less mathematical operations than a common mean-value engine-model.

A model that incorporates all these properties is not available to date. The contribution of this work is the development of such a model and its validation against specific, transient measurement data recorded with appropriate equipment.

During development, the following procedure was pursued. Borrowing from phenomenological approaches, physical relations were used to constitute the foundation of the model. They enable the coverage of all qualitative trends including those not explicitly excited during the measurements and ensure a plausible extrapolation behaviour. This base was then combined with the execution speed, the quantitative accuracy and the straightforward identification of empirical models, but without requiring the large amount of measurement data and relying on stationary measurements only. The following three principles were applied in order to achieve this synthesis.

i) Identification of the most important physical relations, omitting negligible effects and introducing simplified but physically motivated relations where needed;

ii) Use of a setpoint-relative formulation as proposed in [65, 90, 45], providing quantitative accuracy while enabling further simplifications;

iii) Adherence to model inputs that are easily obtained from a mean-value engine model [92], and which are extracted from the measured in-cylinder pressure during the identification phase to assure physical plausibility.

In a setpoint-relative formulation, measured reference-values y_{SP} of the model output are utilised to avoid direct prediction of the absolute values y by an ab-initio model. The inputs $\boldsymbol{u}$ leading to these reference-outputs are termed "setpoints" $\boldsymbol{u}_{\mathrm{SP}}$. A setpoint-relative model only has to reproduce the change of the output relative to its reference-value for deviations from these setpoints,

$$y = \mathscr{F}(\boldsymbol{u}) \quad \longrightarrow \quad \frac{y}{y_{\mathrm{SP}}} = \frac{\tilde{\mathscr{F}}(\boldsymbol{u})}{\tilde{\mathscr{F}}(\boldsymbol{u}_{\mathrm{SP}})} \tag{2.84}$$

Tilde is used to designate a setpoint-relative function or intermediate quantity. For an engine, it is natural to use the engine operating-point, defined by its rotational speed and the injected fuel-mass, as scheduling parameters for the setpoints. Therefore, a setpoint-relative function $\tilde{\mathscr{F}}$ only has to describe the relative behavior of the underlying physics for constant engine speed and load.

To assess the achievement of all objectives listed above, the model has been validated using transient measurements on different engines. Thereby, combined variations in all relevant inputs on several predefined test cases were performed. Explicitly varying the static inputs

during transient operation also excites the cross influences between those and the engine dynamics.

In this work, boost level (controlled by a variable-geometry turbine), combustion timing and common-rail pressure are chosen as the controls relevant for the dynamic optimisation. Thus, the model has to be able to predict their influence on the NO_x emissions. External EGR is inherently captured by the model since it is sensitive to the gas composition and the temperature of the cylinder charge as will be illustrated in the next section.

Assumptions The model relies on the following main simplifications and assumptions.

A1) No pilot or post injections. Post injections are found to have only marginal influence on NO_x emissions [31]. Pilot injections are excluded since they are primarily used to reduce combustion noise at low engine power where the absolute mass of NO_x emissions and thus the potential for reduction is small. In fact, the influence of the pilot-injection strategies applied in the engines under consideration change the NO_x emissions by 3-4% only, which has been experimentally validated on the entire engine operating-range.

A2) Combustion characteristics such as the fraction of premixed combustion, the degree of incomplete combustion or the wall-impingement of the fuel spray do not significantly change within one operating point.[10]

[10] This assumption has been found to hold true except for extreme scenarios, see Sec. 2.5.7 and the paragraph on the "NO_x-bump region" in Sec. 2.1 which is illustrated in Fig. C.8.

A3) Thermal NO-formation dominates other formation mechanisms, is primarily rate controlled (i.e. the equilibrium concentration is never reached) and takes place in the region adjacent to the flame and in the hot post-combustion gases. All these assumptions are widely accepted for classical diesel combustion [169, 59, 60, 43, 188, 102].

A4) The composition of all gas mixtures is sufficiently described by the four most prominent molecular species, namely oxygen (O_2), nitrogen (N_2), water (H_2O) and carbon dioxide (CO_2).

A5) The spatial and temporal distribution of temperature and composition, which quasidimensional models try to capture, may be approximated by considering a single representative packet. This reduction is justified by the similarity of temperature and composition profiles of different fuel packages shown in [67, 68, 180] and is further encouraged by the results presented in [113], where the effect of the number of packages has been investigated.

A6) Combustion and emission formation are treated as quasi-static processes, i.e. they are fully defined by the state and composition of the cylinder charge at intake-valve closing (IVC) and the injection parameters. Moreover, no cycle-to-cycle variations are considered.

2.5.3 Description of the model

The structure of the NO_x-formation model, as it would be embedded in a mean-value engine model, is shown in Fig. 2.35. Of the three control inputs, namely the position of the variable-geometry turbine (u_{VGT}), the start of main injection (φ_{SOI}) and the common-rail pressure (p_{rail}), only the last one has a direct influence on the NO_x formation. The start of injection (SOI) affects the start of combustion (SOC) and

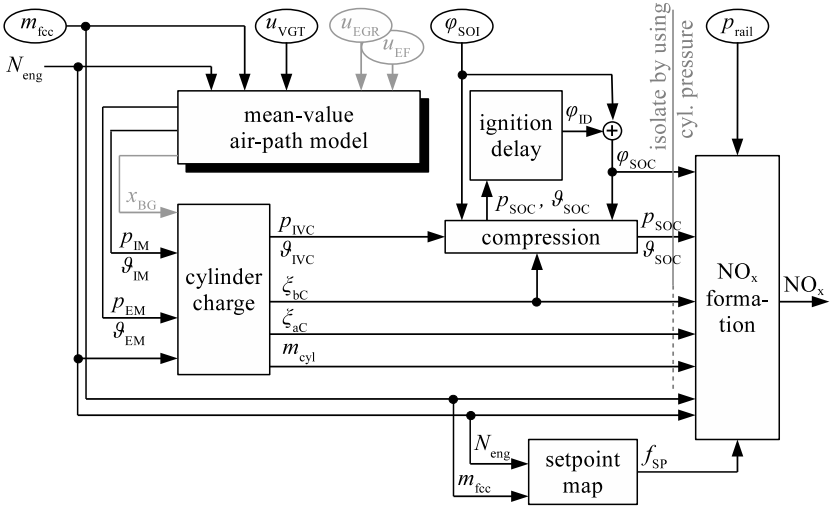

Figure 2.35: Structure of the NO_x model. In this section, only the blocks "setpoint map" and "NO_x formation" are described. The parts in grey are present only in the model for engines with EGR.

u_{VGT} changes the air-path states such as pressures and temperatures, which define the cylinder charge at IVC. Calculating this state from measured signals as described in C.2 and using the measured cylinder pressure enables the isolation of the NO_x-formation model from other submodels as indicated in Fig. 2.35. For engines with EGR and possibly an exhaust flap (EF), the corresponding air-path actuators are inputs to the dynamic air-path model and the burnt-gas fraction in the intake manifold is an additional input to the submodel for the cylinder charge. As mentioned above, the influence of EGR is not analysed explicitly in this section but is deferred to Sec. 2.5.6 below.

The start of combustion as well as the pressure and temperature in the cylinder at that time are intermediate inputs to the model. These quantities can be either calculated from the cylinder pressure, e.g. dur-

ing identification or when the model is applied online on an engine-control unit, or by models for the cylinder charge, compression and ignition delay. These processes are extensively discussed in literature and for each, models of various levels of detail are available. A good starting point are compendia such as [92, 102]. It should be noted that the required complexity of these models changes with the complexity of a given engine, e.g. its speed range and cylinder configuration, the presence of swirl control, variable valve-timing, and others. With these submodels available, the NO_x model can be used on an engine controller when online measurement of the in-cylinder pressure is not available, or embedded in a mean-value engine model. The latter enables its use in engine simulators and for controller development, model-based engine calibration or optimal control as will be shown in Chapters 4 and 5 of this thesis.

The remainder of this section is structured as follows. First, the basic structure of the model is derived, then the equations describing the individual physical processes are stated and the setpoint-relative formulation is developed. Finally, the steps required for model evaluation are summarised to provide an overview of the final model structure and the equations involved.

Basic structure A single global reaction is used as supported by [169, 14, 15, 133, 43, 102, 136],

$$\frac{1}{2}\,\mathrm{N_2} + \frac{x}{2}\,\mathrm{O_2} \rightleftharpoons (2-x)\mathrm{NO} + (x-1)\mathrm{NO_2} \equiv \mathrm{NO}_x, \tag{2.85}$$

with the reaction rates modelled as

$$r_{\mathrm{fw}} = k_{\mathrm{fw}}(\vartheta) \cdot [\mathrm{N_2}]^{\frac{1}{2}} \cdot [\mathrm{O_2}]^{\frac{x}{2}}, \quad r_{\mathrm{bw}} = k_{\mathrm{bw}}(\vartheta) \cdot [\mathrm{NO}_x]. \tag{2.86}$$

The formation rate, i.e. the net reaction rate times the volume participating in the reaction, is integrated over one combustion cycle, yielding the amount of NO_x produced:

$$\frac{n_{NO_x}}{6 \cdot N_{eng}} = \int_{\varphi_{SOC}}^{\varphi_{EVO}} V_{reac}(\varphi) \cdot (r_{fw}(\varphi) - r_{bw}(\varphi))\, d\varphi \tag{2.87}$$

The engine speed N_{eng}, which is assumed to remain constant over any one cycle, is omitted from here on since it eventually cancels in the setpoint-relative formulation.

A crank-angle resolved solution of interconnected reaction kinetics has to be avoided in order to provide a sufficiently high execution speed. Moreover, according to assumption A5, neither local and time-resolved values for the concentrations nor the corresponding reaction volume are available. These quantities need to be approximated by representative scalar values, reducing (2.87) to

$$\begin{aligned} \tilde{n}_{NO_x} &= \tilde{V}_{reac} \int_{\varphi_{SOC}}^{\varphi_{EVO}} \left(k_{fw}(\vartheta)[N_2]^{\frac{1}{2}}[O_2]^{\frac{x}{2}} - k_{bw}(\vartheta)[NO_x] \right) d\varphi \\ &= \tilde{V}_{reac} \cdot [N_2]^{\frac{1}{2}}[O_2]^{\frac{x}{2}} \cdot \int_{\varphi_{SOC}}^{\varphi_{EVO}} k_{fw}(\vartheta) \cdot \ldots \\ &\quad \ldots \left(1 - \frac{k_{bw}}{k_{fw}}(\vartheta) \cdot \frac{[NO_x]}{[N_2]^{\frac{1}{2}} \cdot [O_2]^{\frac{x}{2}}} \right) d\varphi. \end{aligned} \tag{2.88}$$

By assuming that the ratio of reaction constants does not significantly change over the temperature range, the second term of the integrand becomes constant:

$$x_{fb} := 1 - \frac{k_{bw}}{k_{fw}} \cdot \frac{[NO_x]}{[N_2]^{\frac{1}{2}} \cdot [O_2]^{\frac{x}{2}}}. \tag{2.89}$$

This factor accounts for the backward reaction by estimating its intensity relative to the forward reaction.

In order to calculate x_{fb}, the concentrations are approximated by their values at the end of the combustion cycle. The remaining amounts of nitrogen and oxygen are calculated by the combustion stoichiometry, cf. Appendix B. An estimate for the amount of NO_x after the combustion is obtained by considering the forward reaction only. The terms in Eq. (2.88) not included in x_{fb}, omitting the concentrations, read

$$\tilde{n}_{\mathrm{NO_x,fw}} = \tilde{V}_{\mathrm{reac}} \cdot \int_{\varphi_{\mathrm{SOC}}}^{\varphi_{\mathrm{SOC}}+\Delta\varphi} k_{\mathrm{fw}}(\vartheta)\, d\varphi. \tag{2.90}$$

The concentrations are neglected for several reasons. First, the effect of higher flame temperature outweighs the higher availability of oxygen when its concentration is increased [152, 141, 63]. This finding is supported by the widely used correlation between the flame temperature and NO_x emissions described in the next paragraph and also holds for the measurements of EGR variations at hand. Second, a representative scalar value cannot be defined in the simple model structure used here. The concentrations remaining after combustion, considered in the backward-reaction compensation in Eq. (2.89), provide an estimate for the general availability of the corresponding species during the cycle and thus also account for the marginal influence of nitrogen and oxygen concentrations. Third, the concentrations are proportional to the cylinder-charge density, which is compensated by the reaction volume as detailed later.

The integration is performed over a constant interval of $\Delta\varphi = 30°$ crank angle, starting at SOC. The interval is chosen such that, in every case, the reaction constant becomes negligibly small before the end of the integration process. Using an equal discretisation of 1° crank angle leads to a simple sum for the integration.

Physical phenomena The most crucial quantity in the model is the temperature of the single representative package since it defines the reaction constant $k_{fw}(\vartheta)$. Its initial value is chosen to be the adiabatic flame-temperature (AFT) ϑ_{af}. In [152] and in the literature referred to in the introductory section of [154], a simple Arrhenius-like function of the form $\exp(-\vartheta_A/\vartheta_{af})$ is shown to correlate changes of the intake temperature and the gas composition to the NO_x emissions. A constant activation temperature ϑ_A is used, and the AFT is calculated from the cylinder-charge composition before combustion, ξ_{bC}, and the temperature at start of combustion ϑ_{SOC}. Neglecting dissociation effects and linearising the thermodynamic properties in the two relevant temperature regions enables an explicit formulation of the AFT, see Appendix B.2 for details.

Two effects define the evolution of the temperature of the representative package over the cycle: Compression and expansion of the cylinder charge and heat losses from the formation zone to the cooler surrounding gas. The former is accounted for by assuming polytropic compression and expansion, whereas the latter is expressed by a temperature reduction due to heat losses $\Delta\vartheta_{hl}$,

$$k_{fw}(\varphi) = \exp\left(-\frac{\vartheta_A}{\vartheta_{reac}(\varphi)}\right), \tag{2.91}$$

$$\vartheta_{reac}(\varphi) = \vartheta_{af}(\xi_{bC}, \vartheta_{SOC}) \cdot \left(\frac{V_{cyl}(\varphi_{SOC})}{V_{cyl}(\varphi)}\right)^{\gamma(\varphi_{SOC})-1} - \Delta\vartheta_{hl}. \tag{2.92}$$

Here, the isentropic exponent has been chosen to be a linear function of the combustion timing relative to top dead-centre,

$$\gamma = \gamma_0 + \gamma_1 \cdot (\varphi_{SOC} - 360^\circ). \tag{2.93}$$

In order to estimate the heat losses, a characteristic temperature difference is required. The AFT is representative for the formation zone. For its surroundings, it is assumed that the heat introduced by the combustion of the fuel is absorbed by a deliberately chosen fraction of the cylinder mass, yielding the temperature

$$\vartheta_{\mathrm{surr}} = \vartheta_{\mathrm{SOC}} + \frac{H_{\mathrm{l}} \cdot m_{\mathrm{fcc}}}{c_{\mathrm{p}} \cdot 0.5 \cdot m_{\mathrm{cyl}}}. \tag{2.94}$$

For the specific heat c_{p}, a value of $1150\,\mathrm{J/kg{\cdot}K}$ is used, which corresponds to standard dry air at 1000 K. The temperature loss due to the heat transfer is then given by fitting a lumped parameter,

$$\Delta\vartheta_{\mathrm{hl}} = k_{\mathrm{hl}} \cdot (\vartheta_{\mathrm{af}} - \vartheta_{\mathrm{surr}}) \tag{2.95}$$

The amount of fuel injected determines the intensity of the heat transfers and thus the parameter k_{hl}. If the injection quantity exceeds a certain amount, the representative fuel package is embedded between previously burned and still burning packets. Only the first and the last portions of the burning fuel spray border on the surroundings. Thus, the heat-transfer parameter k_{hl} tends towards a small or even slightly negative value. When only a small amount of fuel is injected, the situation changes: In the notional case of only one small packet being burned, the formation zone is fully surrounded by cool unburned gas and thus is excessively cooled. Identifying k_{hl} individually for each operating point as explained below in Sec. 2.5.4 clearly discloses this behaviour as shown in the left-hand plot in Fig. 2.36. This characteristic can be represented by a threshold with a quadratic rise:

$$k_{\mathrm{hl}} = k_{\mathrm{hl},0} + k_{\mathrm{hl},1} \cdot \max\{0, m_{\mathrm{fcc,thr}} - m_{\mathrm{fcc}}\}^2. \tag{2.96}$$

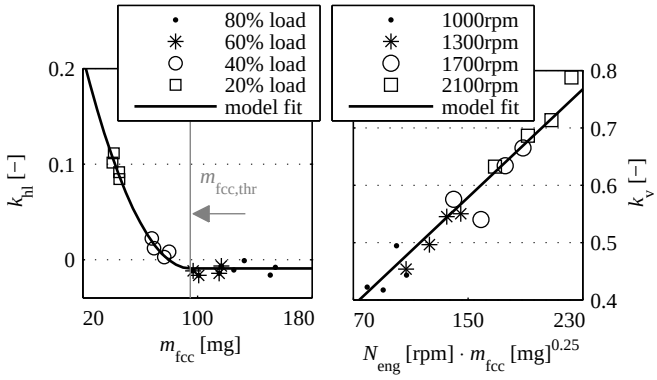

Figure 2.36: Heat-transfer parameter (left) and exponent of the injection pressure-difference (right), each fitted individually for each operating point and the model equation using the final parameters, data from engine A.

Given the injection quantity as it is the case for a setpoint-relative model, the volume involved in the combustion is mainly defined by the pressure difference across the injection nozzles. Several effects constitute this behaviour. First, a higher injection pressure results in a greater length of spray-penetration since the injection speed is roughly proportional to the square root of the pressure difference. Second, the spray consists of smaller droplets, which provide a higher surface-to-volume ratio and thus a larger flame surface. Both effects are conceptually described in [98]. Finally, the effect of a larger flame surface, accompanied by the turbulence introduced by the spray, also leads to a faster combustion of the fuel. The increase in combustion speed, which is roughly proportional to the injection pressure [96], yields a higher volume at high temperatures. The compression of the injection spray due to the cylinder-charge density is omitted since it is compensated by the accompanied increase in concentrations of the species relevant for the reaction. The notion is that when increasing the density by a factor

of two, the probability of two molecules meeting in a given volume doubles, but at the same time the available volume is halved.

Again, a scalar representative value has to be found since neither crank angle nor spatially resolved calculations of the spray development are performed. All the above considerations lead to the following expression for the formation volume:

$$\tilde{V}_{\mathrm{reac}} = \left[p_{\mathrm{rail}} - \left(p_{\mathrm{cyl}}(\varphi_{\mathrm{SOC}}) + k_{\mathrm{p}} \cdot m_{\mathrm{fcc}} \right) \right]^{k_{\mathrm{v}}} \tag{2.97}$$

$$k_{\mathrm{v}} = k_{\mathrm{v},0} + k_{\mathrm{v},1} \cdot N_{\mathrm{eng}} \cdot m_{\mathrm{fcc}}^{0.25}. \tag{2.98}$$

The pressure opposing the rail pressure during injection is approximated by the cylinder pressure at SOC plus a term representing the pressure rise due to combustion. A higher engine speed increases swirl as well as squish flows. Both interact with the fuel spray by introducing additional turbulence and thus enhancing spray break-up as well as by deflecting the spray [102]. The influence of the injection pressure-difference is therefore expected to depend on the engine speed. Also a small dependency on the injection quantity is found since it defines the injection duration and thus alters the relative impact of the aforementioned effects.

The backward reaction becomes perceivable when the oxygen concentration in the formation zone approaches the scale of the NO_x concentration. Assuming a constant NO_2 fraction of 20% yields a value of $x = 1.2$ in Eq. (2.85). By expressing the concentrations by their amounts and volumes, Eq. (2.89) reads

$$x_{\mathrm{fb}} = 1 - \underbrace{\frac{k_{\mathrm{bw}}}{k_{\mathrm{fw}}} \cdot \frac{V_{\mathrm{N_2}}^{0.5} \cdot V_{\mathrm{O_2}}^{0.6}}{V_{\mathrm{NO}_x}}}_{=:k_{\mathrm{fb}}} \cdot \frac{n_{\mathrm{NO}_x,\mathrm{fw}}}{n_{\mathrm{N_2}}^{0.5} \cdot n_{\mathrm{O_2}}^{0.6}}. \tag{2.99}$$

The amounts of oxygen and nitrogen remaining in the cylinder after complete combustion and the amount of NO_x predicted by the forward reaction, as explained in the preceding section, are used as representative values. In the cylinder, all of the NO_x is concentrated in the formation region, whereas the oxygen and nitrogen are approximately equally distributed throughout the unburned gas and the full cylinder volume, respectively. A representative value for the ratio of these volumes and the reaction constants is introduced in the form of the model parameter k_{fb}.

Setpoint-relative formulation Deriving the setpoint-relative formulation of the model leads to the final equations. Note that throughout this and the next section, the notation $n := n_{NO_x}$ is used for better readability. The model has to predict the relative factor correlating the actual amount produced by the forward reaction to the corresponding setpoint value:

$$\frac{\tilde{n}_{fw}}{\tilde{n}_{fw,SP}} \approx \frac{n_{fw}}{n_{fw,SP}} \approx \frac{n_{fw}}{n_{SP}/x_{fb,SP}} \tag{2.100}$$

The setpoint value is calculated from the measured NO_x emissions n_{SP} and the setpoint value for the back-reaction compensation, $x_{fb,SP}$. Rearranging (2.100) yields

$$n_{fw} = \tilde{n}_{fw} \cdot \frac{n_{SP}}{\tilde{n}_{fw,SP} \cdot x_{fb,SP}} =: \tilde{n}_{fw} \cdot f_{SP}, \tag{2.101}$$

indicating that only one reference value f_{SP} has to be precalculated and stored. Using the resulting n_{fw}, the actual factor x_{fb} can be evaluated according to Eq. (2.99), yielding the model prediction for the amount of

NO_x produced per combustion cycle:

$$n = n_{\mathrm{fw}} \cdot x_{\mathrm{fb}} = n_{\mathrm{fw}} \cdot \left(1 - k_{\mathrm{fb}} \cdot \frac{n_{\mathrm{fw}}}{n_{\mathrm{N_2}}^{0.5} \cdot n_{\mathrm{O_2}}^{0.6}}\right). \tag{2.102}$$

The reference value f_{SP} is precalculated on a fine-sampled engine map, using setpoint values for the control inputs. Inserting (2.90) yields

$$f_{\mathrm{SP}} = \frac{n_{\mathrm{SP}}}{x_{\mathrm{fb,SP}}} \cdot \left(\tilde{V}_{\mathrm{reac,SP}} \cdot \int_{\varphi_{\mathrm{SOC}}}^{\varphi_{\mathrm{SOC}}+\Delta\varphi} k_{\mathrm{fw}}(\vartheta_{\mathrm{reac,SP}})\, d\varphi\right)^{-1}. \tag{2.103}$$

Thereby, $x_{\mathrm{fb,SP}}$ is implicitly given by (2.99): Setting $n_{\mathrm{fw}} = n_{\mathrm{SP}} / x_{\mathrm{fb,SP}}$ results in

$$x_{\mathrm{fb,SP}} = 1 - \frac{1}{x_{\mathrm{fb,SP}}} \cdot \frac{k_{\mathrm{fb}} \cdot n_{\mathrm{SP}}}{n_{\mathrm{N_2,SP}}^{0.5} \cdot n_{\mathrm{O_2,SP}}^{0.6}}, \tag{2.104}$$

which is a quadratic equation with the solution

$$x_{\mathrm{fb,SP}} = \frac{1}{2} \cdot \left(1 + \sqrt{1 - 4 \cdot \frac{k_{\mathrm{fb}} \cdot n_{\mathrm{SP}}}{n_{\mathrm{N_2,SP}}^{0.5} \cdot n_{\mathrm{O_2,SP}}^{0.6}}}\right). \tag{2.105}$$

Model summary During model evaluation, first the amount of NO_x produced by the forward reaction, n_{fw}, is calculated. To do so, its setpoint-relative representation, $\tilde{n}_{\mathrm{fw}}$, defined by Eq. (2.90) is evaluated, using Eq. (2.91) to (2.98), and related to the pre-calculated setpoint-value f_{SP} according to Eq. (2.101). The resulting absolute quantity is then used to compensate for the backward reaction as shown in Eq. (2.102), yielding the final model output n. The reference values, from which the lookup map for f_{SP} is constructed, are calculated by successive evaluation of Eq. (2.105) and (2.103) for each operating point.

Table 2.1: Parameters of the NO_x model, associated physical phenomena, and initial guesses.

Symbol	*Unit*	*Associated physical phenomena*	*Initial guess*
ϑ_A	K	reaction kinetics, gas composition	34300 [102]
γ_0	-	polytropic compression and expansion	1.4 (SDA)
γ_1	$1/deg$	change of the polytropic exponent over the high-pressure cycle	0[b]
k_{fb}	-	ratio of reaction constants and volumes	1
k_p	Pa/kg	pressure increase due to combustion	$1 \cdot 10^{11}$[c]
$k_{v,0}$	-	volume of spray, combustion rate	1 [96]
$k_{v,1}$	$\frac{min}{kg^{0.25}}$	spray deflection, turbulence, timescale	0[b]
$k_{hl,0}$	-	heat transfer at high injection quantities	0[b]
$k_{hl,1}$	$1/kg^2$	change of heat transfer for smaller injection quantities	0[b]
$m_{fcc,thr}$	kg	threshold for heat-transfer parameter	0[b]

[a] fitted using data of engine C

[b] during automatic identification, the initial value is estimated by optimising the base parameter individually in each operating point and fitting the corresponding equation to this data

[c] estimated from $c_p \cdot m \cdot \Delta\vartheta = m_{fcc} \cdot H_l$ and $\Delta p \cdot V = m \cdot R \cdot \Delta\vartheta$ using c_p and R of standard dry air (SDA) and the compression volume for V; the mass m cancels

All parameters appearing in the NO_x-formation model, the physical phenomena they represent and initial guesses are summarised in Table 2.1.

2.5.4 Parameter identification

The identification procedure for the model parameters is fully automatic. This automatisation is made possible by the availability of isolated variations on operating points spanning the full operating range of the engine. Recall the following terms concerning the measurement data. SOI, rail-pressure and boost variations denote measurements

where only the respective input has been varied and the others held constant at their setpoint value. These data as a whole is called "single variations" in contrast to "cross variations", where multiple inputs were varied at the same time. "Two-fold" and "three-fold" thereby indicates whether only two or all three inputs were varied simultaneously.

Each step of the identification procedure consists of a nonlinear minimisation. Any nonlinear optimisation algorithm, such as the one implemented in the MATLAB-function `fminsearch` may be used. As objective function, except for steps 3c and 4c, the equally weighted sum of squared relative errors in predicted NO_x is used. For each consecutive step, the parameter set from the previous step is taken as initial value, such that a cascaded and partly iterative identification results. This structure ensures a reliable and reproducible identification exploiting the underlying physical relations wherever possible. Note that only single variations, i.e. no cross variations, are used for the identification. The individual steps are detailed next.

1. (a) Initialise the parameter set with the initial guesses listed in Table 2.1.

   (b) Fit a reduced parameter set to the sparse data available for engine C. Only the parameters ϑ_A, γ_0, γ_1, $k_{v,0}$ and $k_{v,1}$ are used since the available variations do not excite the other parameters.[11]

   (c) Use the resulting value for ϑ_A as a constant for engines not equipped with EGR such as the two at hand, reset the other parameters to initial guesses.

[11] If no EGR-equipped engine is available at all, the initial guess taken from literature may be used for ϑ_A. Alternatively, the concentration of some species could be explicitly varied by adding calibration gas to the intake air [152]. The adaptation of this identification procedure for engines with EGR is described in Sec. 2.5.6.

2. Identify the parameters γ_0, γ_1 in (2.93) and k_{fb} in (2.102) using the SOI variations.

3. The following sub-steps are illustrated in the left-hand plot of Fig. 2.36.

   (a) Fit k_{hl} in (2.95) individually for all operating points using the boost variations.

   (b) Select $m_{\mathrm{fcc,thr}}$ at the apparent turning point; this parameter is not changed later.

   (c) Obtain initial values for $k_{\mathrm{hl},0}$ and $k_{\mathrm{hl},1}$ by nonlinear minimisation of the sum of squared errors between the fit (2.96) and the operating-point individual values for k_{hl} obtained in 3a.

   (d) Fine-tune $k_{\mathrm{hl},0}$ and $k_{\mathrm{hl},1}$ using the boost variations of all operating points at once.

4. The following sub-steps are illustrated in the right-hand plot of Fig. 2.36.

   (a) Fit k_{p} and k_{v} in (2.97) using the rail-pressure variations to obtain a plausible value for k_{p}.

   (b) Fit k_{v} individually for each operating point using the rail-pressure variations.

   (c) Obtain initial values for $k_{\mathrm{v},0}$ and $k_{\mathrm{v},1}$ by linear least-squares minimisation of the error between the fit (2.98) and the operating-point individual values for k_{v} obtained in 4b.

   (d) Fine-tune $k_{\mathrm{v},0}$ and $k_{\mathrm{v},1}$ along with k_{p} using the rail-pressure variations of all operating points at once.

5. Fine-tune all parameters except for ϑ_{A} and $m_{\mathrm{fcc,thr}}$ using all single variations at once.

Setpoint maps and preliminary results The average magnitude of the relative errors over all stationary variations, including the two- and three-fold cross-variations not used for identification, is 2.63% for engine A and 3.13% for engine B without EGR, respectively. Figure 2.37 shows plots of the modelled versus measured NO_x factors, i.e. the actual values resulting from variations in the inputs compared to the measured setpoint value. Applying the parameter set of one engine to the data of the other engine leads to an average error-magnitude of only 7% in both cases. Including the cross variations in the identification procedure hardly improves the model quality and merely leads to marginal changes in the resulting parameters. On the other hand, using the largest single variations only during identification increases the error by 0.15% for both engines while hardly affecting the parameter values at all. Therefore, the identification of the model requires merely six extra measurements in a handful of operating points when obtaining the fine-sampled reference map from the engine test-bench.

The most relevant setpoint maps are depicted in Fig. 2.38 for engine A. As expected, the forward reaction tends towards zero for low loads. The backward-reaction compensation only becomes perceivable for operating regions with inherently low oxygen availability, i.e. at low air-to-fuel ratios combined with high fuel-specific NO_x emissions. Besides high loads, low engine speeds also encourage this effect due to low boost levels being caused by a small enthalpy flow in the exhaust gas. Consequently, the low-speed but high-load region is affected most. The one reference map that has to be stored and is required for model evaluation, defined by Eq. (2.103), is shown for completeness.

In Fig. 2.39, the model output during the transient cycles is compared to the measured NO_x emissions. For each of the three cycles, the reference, using setpoint control-inputs, and the variations leading to highest (lowest) emissions are shown. The latter are defined by both

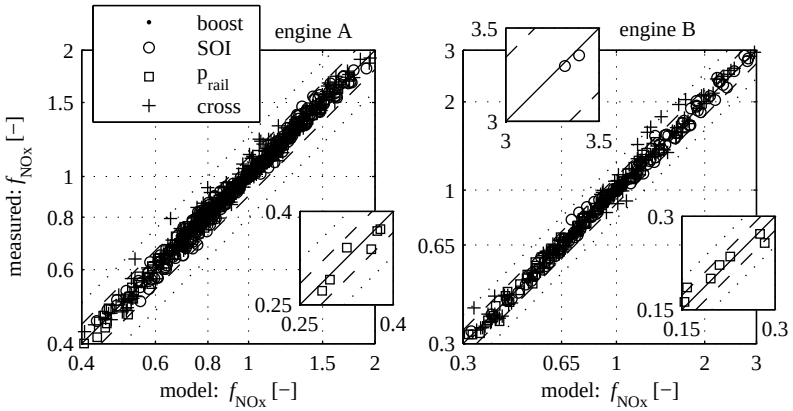

Figure 2.37: Stationary results of the NO_x model: relative change of the NO_x emissions predicted by the model compared to the measurement data, all operating points and variations, lines for 10% and 20% relative error shown. Only data without EGR shown for engine B.

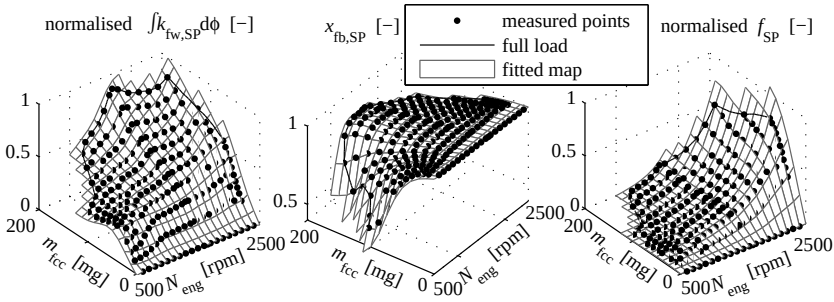

Figure 2.38: Setpoint maps of the NO_x model for engine A: integral of forward reaction-constant (normalised, left), backward-reaction compensation-factor (center) and reference value to be stored (normalised, right); normalised values shown since maps are reference for a setpoint-relative model.

higher (lower) rail pressure combined with early (late) injection. In addition, the corresponding intermediate variations are shown. Over all variations performed, the average magnitude of the relative error amounts to 2.5%, and the largest deviations never exceed 10% for the acceleration and constant-speed cycles. The figures for the constant-torque cycle are 4.5% and 18%, respectively. This worse accuracy is attributed to the fast changes in engine speed and thus in air and exhaust mass-flows during this cycle, as shown in Fig. 2.3 d). The transient compensation of the temperature measurements as described in section 2.1 introduces uncertainties by neglecting the flow dependency of the time constant and the heat exchanges with the manifold walls [168]. Furthermore, the gas inertia may become relevant for the compensation of the air mass-flow.

The NO_x emissions and thus the model are sensitive to errors in measured temperatures and air flow [14, 15]. The term sensitivity here denotes a factor relating a relative deviation in an input to the resulting relative change of the output. An analysis of the operating-range averaged sensitivities from measured quantities to the predicted NO_x emissions yields values of 2.5 and 0.5 for the temperatures in the intake and exhaust manifolds, respectively. Also, the air mass-flow and the boost-pressure measurements exhibit sensitivities of -1 and -0.75, respectively. Sensitivities towards intermediate quantities to be calculated from the submodels shown in Fig. 2.35 indicate the required accuracy of these models. The most critical quantity is the cylinder pressure at SOC since it defines the corresponding temperature by the ideal gas law. It exhibits a sensitivity of 3.5. With values of -0.5 and -0.3, respectively, the ignition delay and residual-gas fractions are less sensitive, and the influence of errors in the calculated cylinder mass varies between -1 and 1 over the engine operating-range.

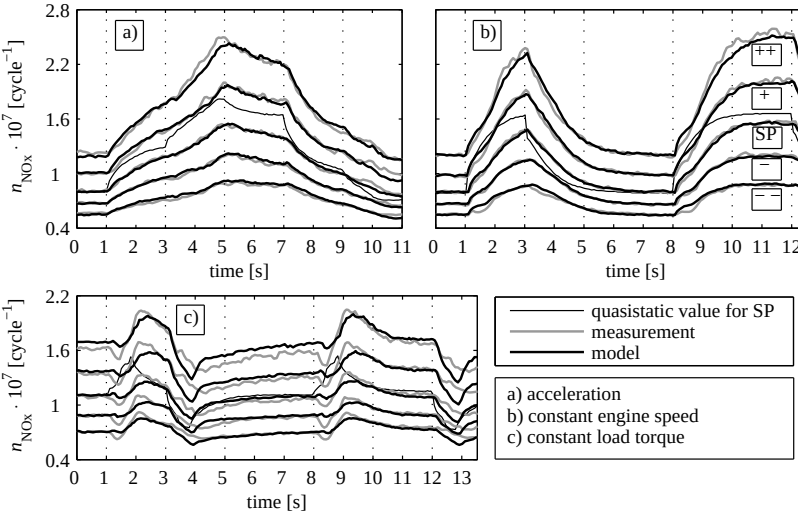

Figure 2.39: Transient results, engine A: Amount of NO_x produced per stroke during transient test-cycles. Reference with setpoint values for control inputs, variations in rail pressure and injection timing: $\pm\{125\text{bar}, 2°\text{CA}\}$ [+, -] and $\pm\{250\text{bar}, 4°\text{CA}\}$ [+ +, - -]

2.5.5 Extension and analysis of the model

In this section, the representation of the single lookup map of the NO_x model by a polynomial is presented. Furthermore, this map is compared to a typical map of an empirical model for the NO_x emissions. Finally, a sensitivity analysis of the NO_x model, including the cylinder submodels, indicates which quantities of the air path are critical for the accuracy of the model prediction.

Polynomial representation of the lookup map The reference value f_{SP} is stored as a lookup map over engine speed and injection quantity. Since it describes a physical process, it can be expected to – and in fact does – lie on a smooth surface, see Fig. 2.40 a). The simple shape of the

latter suggests that it may be well approximated by a polynomial. Doing so reduces the number of parameters of the NO_x model[12], enables the identification of all parameters without requiring a nominal map and enables automatic differentiation to be applied.

The basically exponential shape of the lookup map is captured by approximating its logarithm by a polynomial. The arguments, namely engine speed and fuel quantity, are scaled such as to exhibit similar magnitudes. A simpler notation is used to improve readability: $N := N_{\mathrm{eng}}/1600$, $m := m_{\mathrm{fcc}} \cdot 10^4$. A close fit can be achieved by using the following regressors.

$$\begin{aligned}\log_e(f_{\mathrm{SP}}) \approx f_0 + f_1 \cdot N + f_2 \cdot N^2 + f_3 \cdot m + f_4 \cdot m^2 + \\ f_5 \cdot m^3 + f_6 \cdot N \cdot m + f_7 \cdot N \cdot m^2 + f_8 \cdot N^2 \cdot m.\end{aligned} \quad (2.106)$$

At zero load, i.e. no fuel injection, no NO_x is produced. However, the argument of the exponential function representing the map would be required to approach minus infinity to reproduce this behaviour. To eliminate this mismatch, a linear correction is applied below 5% relative load, which equals a fuel mass of 10 mg injected per cylinder and cycle.

$$n_{\mathrm{lc}} = \begin{cases} n_{\mathrm{mdl}}, & \text{if } m_{\mathrm{fcc}}\ [\mathrm{mg}] \geq 10 \\ n_{\mathrm{mdl}} \cdot \frac{m_{\mathrm{fcc}}\ [\mathrm{mg}]}{10}, & \text{if } m_{\mathrm{fcc}}\ [\mathrm{mg}] < 10. \end{cases} \quad (2.107)$$

The coefficients $f_0 \ldots f_8$ of the polynomial map need to be identified along with the parameters of the NO_x-formation model. To obtain reasonable initial values, the polynomial in Eq. (2.106) is first fitted to the lookup map for f_{SP}. For the NO_x-formation parameters, the values of the original model are used as initial guess. Starting from this initial point, a nonlinear search is performed to fine-tune the poly-

[12] A lookup map has as many parameters as it has grid points.

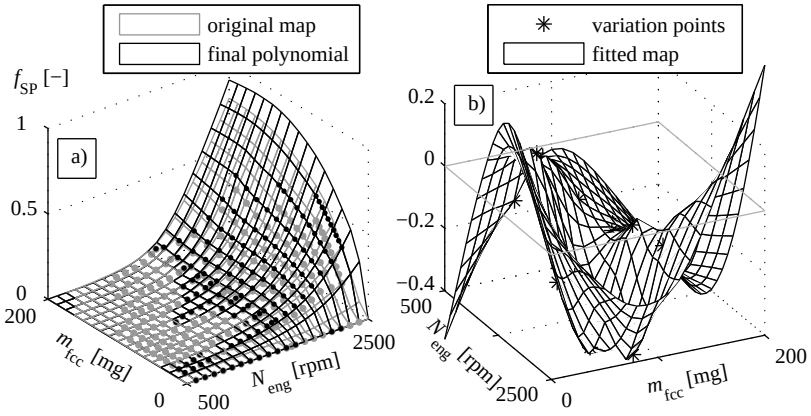

Figure 2.40: a) Original lookup map and polynomial approximation after reidentification along with all parameters of the NO_x-formation model. The linear correction of Eq. (2.107) is applied to the polynomial map. The dots represent the operating-point grid on which the original lookup map was identified. b) Coefficient of the second-order rail-pressure term of the purely quadratic empirical model for the NO_x emissions.

nomial coefficients and the NO_x-formation parameters at once. The polynomial map retains a shape close to that of the original lookup map, see Fig. 2.40 a). Furthermore, the NO_x-formation parameters are only changed marginally by the fine-tuning process. Thus, replacing the lookup map by a polynomial does not impair the physical meaningfulness of the model.

Comparison to an empirical model To illustrate that the smoothness of the reference map is not a general feature of any NO_x model, an empirical model is derived from a second-order Taylor expansion around the setpoints (SP) for each of the 16 variation points. Furthermore, this model is used to assess the prediction capabilities and the execution

speeds of the physics-based model presented here. The model reads

$$n(\boldsymbol{u}) \approx n(\boldsymbol{u}_{\mathrm{SP}}) + \nabla n(\boldsymbol{u}_{\mathrm{SP}})^T \cdot \Delta\boldsymbol{u} + \frac{1}{2}(\Delta\boldsymbol{u})^T \cdot \nabla^2 n(\boldsymbol{u}_{\mathrm{SP}}) \cdot \Delta\boldsymbol{u} \quad (2.108)$$

Here, $\nabla n \in \mathbb{R}^{n_\mathrm{u}}$ is the gradient of the scalar function n w.r.t. the input vector $\boldsymbol{u} \in \mathbb{R}^{n_\mathrm{u}}$, and $\nabla^2 n \in \mathbb{R}^{n_\mathrm{u} \times n_\mathrm{u}}$ is the corresponding (symmetric) Hessian matrix. In each operating point, the following parameters have to be identified.

n_u	reference values for the inputs, $\boldsymbol{u}_{\mathrm{SP}}$
1	reference value for the output $n(\boldsymbol{u}_{\mathrm{SP}})$
$2n_\mathrm{u}$	coefficients for the individual linear and quadratic deviations
$\frac{n_\mathrm{u}(n_\mathrm{u}-1)}{2}$	coefficients for the cross terms of the deviations
$1+\frac{5}{2}n_\mathrm{u}+\frac{1}{2}n_\mathrm{u}^2$	total number of parameters

For the case at hand, with the three inputs φ_{SOI}, p_{rail} and p_{IM} representing the boost level, 13 parameters per operating point result. Note that the reduced data is not sufficient to identify all parameters: It consists of the reference point, yielding the four reference values, and six variations. Thus, only the six coefficients of a second-order approximation of each individual influence, but without the cross-terms, can be identified. Extended to the 16 variation operating points, 208 or 160 parameters result for the full and the reduced approach, respectively.

The main drawback of this kind of model is its lack of extrapolation capability. Even a plausible interpolation cannot be guaranteed. In fact, the maps of the coefficients may become non-smooth and might even exhibit multiple sign changes throughout the operating range of the engine, see Fig. 2.40 b). Furthermore, the empirical model can only predict the effects of the explicitly included inputs, and its range of validity depends on the engine calibration used during the identification.

Table 2.2: Sensitivity of the NO_x emissions w.r.t. the state of the air path and the direct control signals. Rightmost column: average/maximum magnitude of the relative error of the MVEM in stationary operation on the full engine operating-range.

quantity	*change*	*change in NO_x*	*MVEM error*
p_{IM}	1%	−0.2 … 2.6%	1.4 / 6.6%
p_{EM}	1%	−0.4 … 0.2%	1.8 / 8.6%
ϑ_{IM}	1%	1.3 … 4.1%	0.2 / 0.6%
ϑ_{EM}	1%	0.4 … 0.6%	0.6 / 3.2%
φ_{SOI}	0.25 [°] [a]	1.3 … 2.2%	
p_{rail}	12 [bar] [a]	0.8 … 2.1%	
m_{fcc}	1%	−1.5 … 1.1%	0.6 / 2.0%

[a] equals 1% of the common actuator range

Sensitivity analysis By combining the submodels for the in-cylinder processes described in Sec. 2.4 with the NO_x formation model, the sensitivities of the predicted NO_x emissions w.r.t. the air-path quantities can be derived. They are listed in Table 2.2.

The influence of the state of the air path is comparable to the influence of the controls directly affecting the NO_x formation. On one hand, this fact indicates that the air-path control, i.e. the VGT position, also significantly affects the NO_x level, even if no EGR is used. On the other hand, the importance of accurately modelling the temperatures in the intake and exhaust manifolds is indicated. The temperature of the aspirated gas has a large direct effect on the NO_x emissions. The temperature and pressure in the exhaust manifold do not exhibit critically high sensitivities, but indirectly affect the boost level and thus the value of p_{IM} by influencing the enthalpy flow through the turbine.

2.5.6 Analysis of the effect of EGR

The NO_x model presented inherently accounts for the composition and the temperature of the cylinder charge. Therefore, the reduction of the NO_x emissions due to external EGR should be correctly predicted. However, it turns out that at high engine speeds, the model underestimates the NO_x-reduction effect of the EGR. In order to explain this observation, the heat-release rates are analysed for low and high engine speeds.

Figure 2.41 shows the heat-release rates for low and high engine speeds, and at three relative loads. At low engine speed, the EGR does not change the characteristic of the combustion, and the model does account for the later start of combustion due to the longer ignition delay. In contrast, at high engine speed, the EGR reduces the peak of the premixed combustion and also reduces the speed of the subsequent diffusion combustion. At the same time, the ignition delay is hardly affected at all.

This phenomenon can be explained by the turbulence introduced by the higher engine speed. The turbulence reduces the ignition delay by promoting evaporation and mixing of the fuel, rendering the effect of low oxygen availability less relevant. Since the fuel preparation is hardly affected by the EGR, the only remaining influence on the combustion is the lower oxygen availability itself, leading to a slower and thus cooler combustion. However, the NO_x model does not account for such effects.

A simple extension of the model enables the reproduction of this effect. The combustion speed and thus the NO_x formation is sped up by a higher oxygen availability, and this increase is more significant for higher engine speeds. From another point of view, a larger region in the cylinder reaches a temperature sufficiently high for thermal NO_x formation. Thus, by adding an extra factor, the reaction volume described by

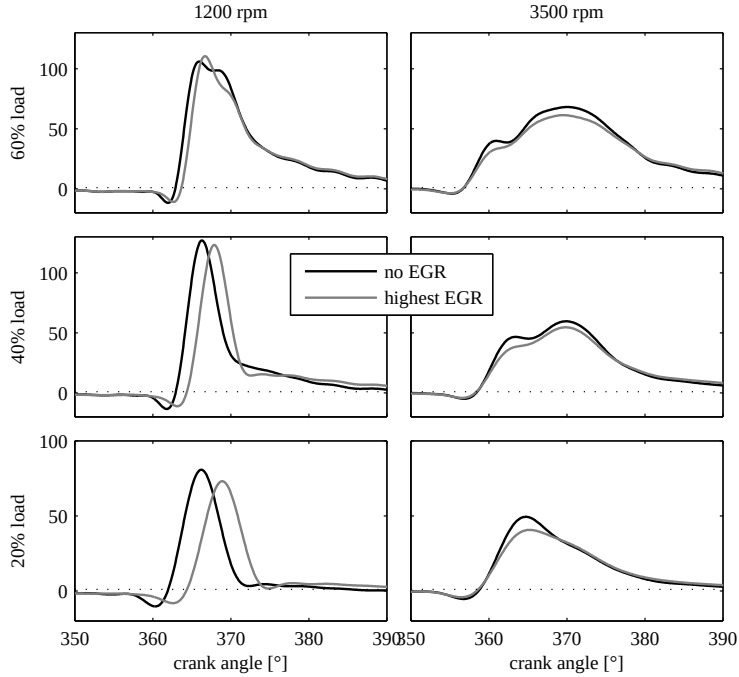

Figure 2.41: The heat-release rate calculated from the cylinder pressure by Eq. (C.3) is shown, in [J/°], demonstrating the influence of the EGR on the combustion for engine B. In all points, the cases without EGR and with maximally achievable EGR are shown, e.g. a fully open EGR valve and a completely closed exhaust flap.

Eq. (2.97) is modified to

$$\tilde{V}_{\text{reac}} \longrightarrow \xi_{\text{O}_2,\text{bC}}^{k_{\text{O}_2} \cdot N_{\text{eng}}/1000} \cdot \tilde{V}_{\text{reac}}. \tag{2.109}$$

The engine speed is normalised to bring the exponent into a suitable order of magnitude. A value of $k_{\text{O}_2} = 1$ is used to initialise the tuning parameter. During the parameter identification, ϑ_{A} and k_{O_2} are held constant and are fine-tuned along with all other parameters in step 5

of the identification procedure described in Sec. 2.5.4. The effect of the EGR, which reduces the NO_x emissions by up to a factor of 18, is predicted by the extended model with an average magnitude of the relative error of 8%.

2.5.7 Limitations and future work

One fundamental assumption of the NO_x model is that the combustion characteristic does not change. It is assumed that a "classical" diesel combustion is maintained throughout the entire operating range and at all plausible actuator settings. Here, "plausible" denotes settings which form a tradeoff between at least two quantities of interest, i.e. they are Pareto-optimal. Points which do not meet this requirement have to be excluded from the identification data. More important, the corresponding regions in the entire actuator space, which is four-dimensional for an engine with EGR, have to be identified. In the optimal control problem, the allowable actuator ranges need to be limited accordingly. This step is cumbersome to be executed by hand due to the high dimensionality of the space to be considered and since the effect of one actuator cannot be isolated from others. An automated identification of such regions during the measurement of the variations could alleviate this problem.

One example of such a region, besides the "NO_x-bump region" described in Sec. 2.1, is the case of a very low boost pressure. Figure 2.42 shows the heat-release rate for low, medium and high boost levels in the four central operating points. Since the rail pressure is lower at the lower load points, the effect of the boost pressure on the ignition delay is more pronounced. In spite of this fact, the combustion always has a significant diffusion part at the higher load points. However, at lower load, the combustion characteristic is substantially changed by the variation of the boost pressure. At 2750 rpm, a typical diesel combustion

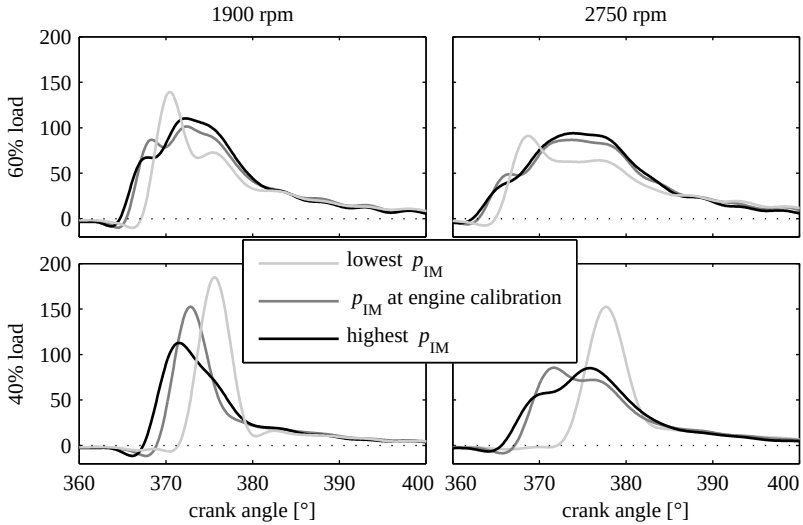

Figure 2.42: The heat-release rate calculated from the cylinder pressure by Eq. (C.3) is shown, in [J/°], demonstrating the influence of extreme variations of the boost pressure on the combustion for engine B. In all points, the largest possible variations are shown along with the settings from the pre-series engine calibration, which is approximately the midpoint in all cases.

is observed at the highest possible pressure. At the calibration point, there is a significantly larger premixed part. Finally, at the lowest boost pressure, an almost purely premixed combustion prevails.

Figure 2.43 shows the NO_x-fuel tradeoff for boost-pressure variations in these four operating points. For both higher load points, there exists some range of the boost pressure where an actual tradeoff between NO_x emissions and fuel consumption is achieved. In this range, lowering the boost pressure leads to a slower and cooler combustion, which is less efficient but also produces less NO_x. The decrease in fuel efficiency for high boost pressures is due to an overproportional increase

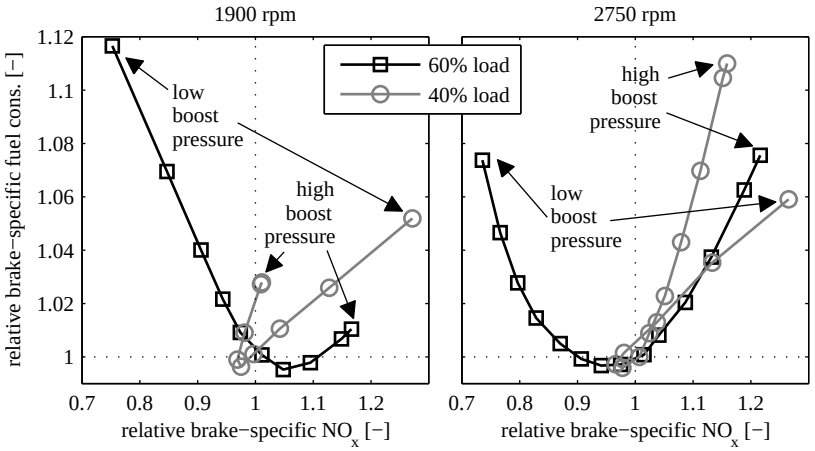

Figure 2.43: NO_x-fuel tradeoff for boost-pressure variations, engine B. Both quantities are related to their values resulting from the actual pre-series engine calibration.

of the exhaust pressure. To achieve these boost levels, the turbocharger is operated far from its design point and thus exhibits a low efficiency.

In contrast, a higher fuel consumption and more NO_x emissions result from too high *and* too low boost pressures at the low-load points. For low pressures, the effect described above leads to a delayed, but fast and hot combustion. The late combuston reduces the efficiency, whereas the fast burn rate results in a large amount of hot post-combustion gases, which are ideal for NO_x formation. The effect is similar to the late-injection "NO_x-bump region", although with a different provenance.

If the model could capture such effects, there would be no need to manually exclude non-Pareto regions from the actuator space during the optimisation. The considerations presented in Sec. 2.5.6 represent a first step towards this direction. However, if varying combustion characteristics are explicitly allowed, or even want to be exploited intentionally,

the NO_x model in its current form is not suitable. Rather than adding more and more corrections, the core of the model had to be adapted to the omission of the main assumptions such as a classical diesel combustion or no pilot injections. Even if all corrections are physically motivated, each of them drives the model further into the direction of an empirical model and potentially impairs the physical soundness of its core concepts of the adiabatic flame temperature, the compression and expansion, the heat losses, and the spray volume.

If a "rebirth" of the model is attempted, at least the following aspects need to be considered and analysed thoroughly.

- Influence of the premixed-combustion fraction, which goes hand in hand with the ignition delay.
- Turbulence and its influence on the spray volume, the ignition delay, and the combustion characteristic in general. All of them possibly exhibit cross couplings to the composition of the cylinder-charge as indicated in Sec. 2.5.6.
- Pilot injections. The main concern should be to accurately identify the way in which they influence the NO_x emissions. Do they produce NO_x themselves or is the main impact due to their influence on the characteristic of the main combustion?

Possibly, certain cross influences could render the generalisation of the model superfluous. For example, the application of pilot injections at low-load operating points, which is applied throughout nowadays, drastically reduces or even eliminates the premixed portion of the main combustion. The result is a more uniform combustion characteristic over the operating range of the engine. The effect of the heat introduced by the pilot combustion could be accounted for by a simple model capturing the influence on the pressure and the temperature at the start of the main combustion.

The models for the ignition delay and the combustion efficiency do not account for the influences of varying fuel properties or the composition of the cylinder charge. Recent experimental investigations reveal that biodiesel blends and fuel emulsification change the combustion and emission characteristics of a diesel engine [175, 119, 9, 58]. Furthermore, oxygen enrichment is proposed [126, 125], which changes the composition of the fresh gas. Due to their multiplicative structure, those models can be extended to account for such effects and thus could be used for parametric studies. The original NO_x-formation model itself inherently captures the effects of the charge composition, but the effects of varying fuel properties remain to be included.

Despite its limitations, the model in its current form illustrates the strength of a model that consists of relations that are based on physical principles. It can be identified by a handful of measurements (interpolation), and it is able to predict cross-influences among the inputs or even external influences not included in the data used for its identification (extrapolation). Furthermore, if only the fundamental physical relations are used as the core of the model, while the quantitative accuracy is retained by a setpoint-relative formulation, its execution speed is on par with a purely empirical model. Finally, the NO_x formation in a diesel engine is a complex physical process. The ability of the semi-empirical model to handle such a complex phenomenon indicates that this modelling principle can also be applied to similarly involved processes.

2.6 Soot emissions

The soot emissions are of minor relevance for the engines at hand. The engines are equipped with a diesel particulate filter (DPF), and the soot emissions are low enough such that no active regeneration of the DPF is necessary. The optimisation thus has to maintain a similar level of soot emissions and should not produce any large instantaneous soot peaks. To this end, a simple model is expedient for the prediction of the emission of this species.

The influences of the oxygen availability, represented by the air-to-fuel ratio λ_{AFR}, and of the rail pressure are captured. The model structure is chosen as

$$\tilde{m}_{\mathrm{soot}} := \frac{\overset{*}{m}_{\mathrm{soot}}}{N_{\mathrm{eng}} \cdot m_{\mathrm{fcc}}} = a_0 + \frac{a_1}{\tilde{\Delta} p_{\mathrm{rail}}} + \frac{a_2}{(\tilde{\Delta} p_{\mathrm{rail}})^2} + \frac{a_3}{\Delta\lambda_{\mathrm{AFR}}} + \frac{a_4}{(\Delta\lambda_{\mathrm{AFR}})^2}, \tag{2.110a}$$

with

$$\tilde{\Delta} p_{\mathrm{rail}} = \left(p_{\mathrm{rail}} - p_{\mathrm{rail,SP}}(N_{\mathrm{eng}}, m_{\mathrm{fcc}})\right) - p_{\mathrm{rail,crit}}, \tag{2.110b}$$

$$p_{\mathrm{rail,SP}} = k_0 + k_1 \cdot N_{\mathrm{eng}} + k_2 \cdot m_{\mathrm{fcc}} + k_3 \cdot N_{\mathrm{eng}} \cdot m_{\mathrm{fcc}} + k_4 \cdot N_{\mathrm{eng}}^2 + k_5 \cdot m_{\mathrm{fcc}}^2, \tag{2.110c}$$

$$\Delta\lambda_{\mathrm{AFR}} = \lambda_{\mathrm{AFR}} - 1 \approx \frac{\overset{*}{m}_{\mathrm{air}}}{\overset{*}{m}_{\mathrm{fuel}} \cdot 14.5} - 1. \tag{2.110d}$$

This model is able to qualitatively reproduce the sharp increase of the soot emissions at low AFRs and injection pressures, as shown in Fig. 2.44. The parameter $p_{\mathrm{rail,crit}}$ defines the asymptote of the surface towards low rail pressures, similar to the one that occurs when λ_{AFR} approaches 1. Its value identified for engine A is -1151 bar.

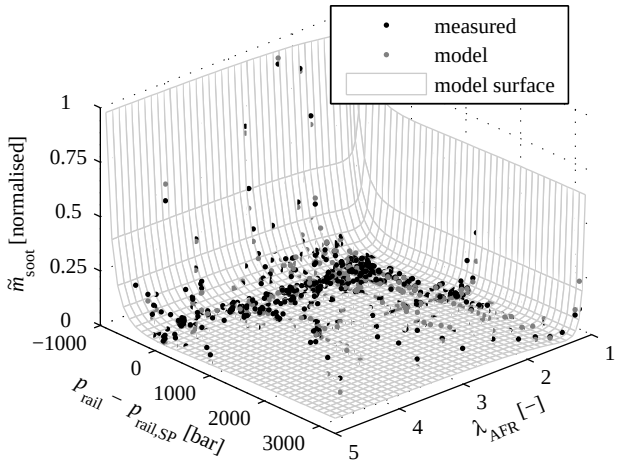

Figure 2.44: The surface of the soot model and the measurement data used for its identification, engine A. The z-axis is normalised for reasons of confidentiality.

As an alternative, a constant or an operating-point dependent lower bound could be imposed on the AFR. In fact, on the ECU the latter is implemented to limit the injection quantity during load increases. For engine B, the simple approach used here is not able to predict the soot emissions. Therefore, the OCP for this engine is adapted accordingly, see Sec. 4.1 below.

2.7 Overall model and validation

The structure of the full engine model for an engine without EGR and without the additional restrictions for the AFM and the ATS is shown in Fig. 2.45. Three model stages are compared to assess the prediction of the NO_x emissions: The isolated NO_x formation, the extension towards mean-value quantities by including the cylinder model, and finally the full mean-value engine model (MVEM). All stages use the polynomial representation of the map in the NO_x-formation model. Except for the first stage, the efficiency model described in Sec. 2.3 is used to calculate the fuel mass. For comparison, the empiric model described in Sec. 2.5.5 is used. Besides the execution speed, the accuracy for stationary and transient engine operation is analysed. Furthermore, the models are evaluated with regard to their ability to predict the influence of hot or cold engine operation.

The execution speed was evaluated on a laptop computer using one processing unit of an Intel® Core™ i7 2760QM CPU with a clock rate of 2.4 GHz. The models are implemented as uncompiled MATLAB (The MathWorks, Inc., Natick, MA, USA) functions. The full MVEM is evaluated along a trajectory resulting from a simulation. The time required for calculating the empiric and the physics-based NO_x models is related to the execution time of the MVEM since both of them require the air-path quantities provided by the MVEM. The execution time for the physics-based model includes the evaluation of the polynomial map and the cylinder model. Table 2.3 summarises the results.

The evaluation of the physics-based model requires approximately twice as much time as that of the empirical model. Compared to the MVEM itself, the physics-based model increases the time for one evaluation by 33%. A forward simulation of the model using MATLAB's ODE15s solver runs around 100 times faster than real-time for a relative integra-

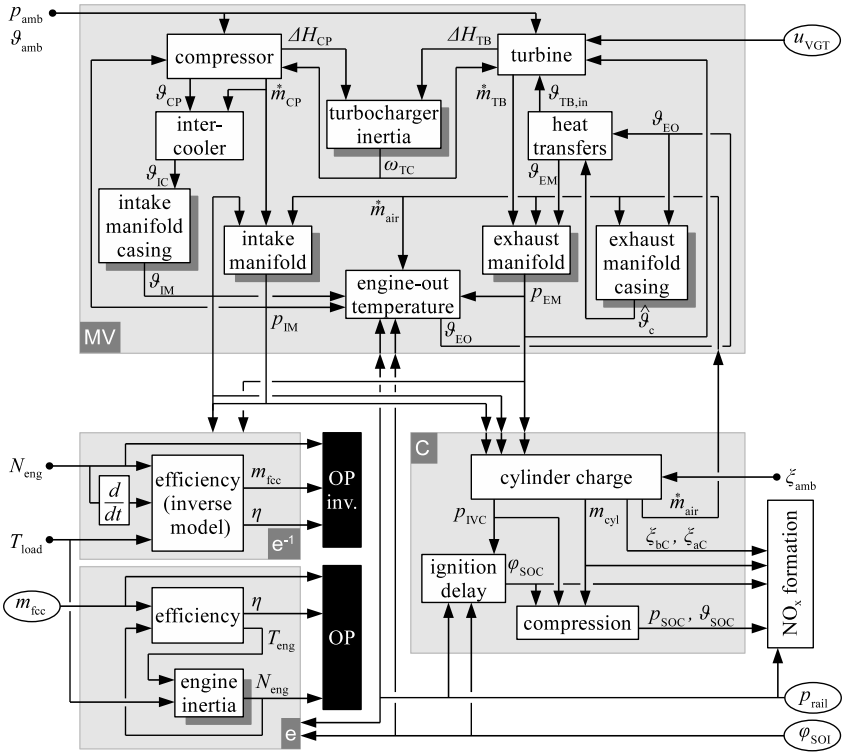

Figure 2.45: Cause-and-effect diagram of the full engine model. Of the two efficiency blocks representing the causal (e) and the inverted formulation (e^{-1}), only one is required. The resulting operating point (OP) is available to all other blocks, which is not explicitly shown to increase the clarity of the diagram. During the transient validation of the NO_x model, first the isolated NO_x-formation model is considered and subsequently, the model is extended to comprise the cylinder model (C) and finally also the mean-value model for the air path (MV).

Table 2.3: Execution speeds for the models. One MVEM evaluation denotes a single calculation of the combustion efficiency and the right-hand sides of the differential equations (2.28), (2.48) and (2.51) defining the MVEM for an engine without EGR.

model	*time/eval.*	*rel. time*	*evals./s*
MVEM	0.150 ms	100%	6'667
MVEM w/ emp. NO_x	0.172 ms	115%	5'797
MVEM w/ phys. NO_x	0.199 ms	133%	5'025

tion tolerance of 10^{-4}. The ODE solver included in the SUNDIALS suite, CVODE, which implements a similar method, is almost twice as fast.

2.7.1 Stationary operation

In Sec. 2.5.4, the stationary accuracy of the model is characterised by the average of the magnitude of the relative error. Here, the standard deviation of the relative error is used as an additional measure. Offset is negligible in all cases and thus is not listed. Table 2.4 shows a comparison of the accuracy figures for stationary operation. All data has been used for the validation. The term "MV quantities" denotes the case in which all signals generated by the MVEM, indicated in Fig. 2.45, are replaced by their measured values.

By definition, the empirical model uses these mean-value quantities as inputs. Its accuracy is slightly higher than that of the physics-based model as long as all data, including the cross-variations, is used for its identification. However, when it is identified using the most extreme single variations only, as in the case of the physics-based model, the empirical model suffers an increase of the relative error of almost a factor of three[13]. Compared to the original model based on the in-

[13] As mentioned in Sec. 2.5.5, the reduced data is not sufficient to identify the cross terms and thus, only quadratic approximations of the individual influences constitute the empirical model in this case.

Table 2.4: Average magnitude and standard deviation of the relative error of the predicted NO_x emissions, stationary operation, engine A. Where nothing else is stated, the reduced data, i.e. only the most extreme single-directed variations, are used during identification.

model stage	*avg(|.|)*	*std. dev.*
original model [18]	2.63%	-
polynomial map (p)	3.13%	4.21%
(p), MV quantities[a]	3.29%	4.51%
(p), MV quantities[b]	2.97%	3.97%
(p), full MVEM[a]	3.57%	5.07%
(p), full MVEM[b]	3.25%	4.52%
empiric, MV quantities[c]	2.56%	4.25%
empiric, MV quantities	7.91%	12.78%
empiric, full MVEM[c]	2.78%	4.60%
empiric, full MVEM	8.13%	13.24%

[a] parameters from stage "polynomial map" used

[b] parameters refitted for stage "MV quantities" used

[c] all data incl. cross-variations used for identification

cylinder pressure, almost no accuracy is lost when the lookup-map is replaced by the polynomial approximation and mean-value quantities are used. However, it is important to fine-tune the parameters of the NO_x-formation model once the required submodels are introduced. This process only slightly changes the parameters, which indicates that the physical meaningfulness of the model is preserved.

2.7.2 Transient operation

Figure 2.47 shows the output of the three model stages during transient engine operation. The driving profile shown in Fig. 2.46 has been recorded for hot and cold engine conditions. The empirical model is

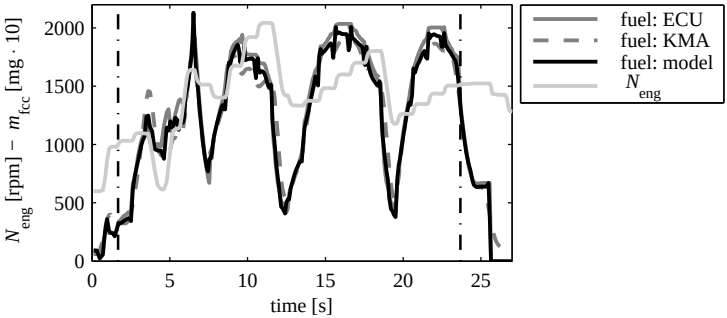

Figure 2.46: Engine-speed and injection-quantity trajectories of the transient test cycle used for the assessment of the full model of engine A. The profile is a segment of the European Transient Cycle.

quite accurate for the hot-engine case. Note that the full version is shown, which is identified including the cross-variations.

As expected, the empirical model is not able to predict the influence of the engine temperature. In contrast, all stages of the physics-based model predict the change in NO_x emissions caused by a reduction of the engine temperature. However, it is crucial for the MVEM to incorporate the thermal dynamics of the manifolds. Especially the thermal dynamics of the intake manifold do not exert any significant influence on the air path as such. Nonetheless, due to the high sensitivity of the NO_x emissions w.r.t. the temperature in the intake manifold, see Table 2.2, they have to be considered when NO_x emissions are to be predicted. This finding confirms the proposition of [156, 157] that the combustion and thus the fuel consumption and emission formation are barely affected at all by the thermal state of the cylinder block itself. Rather, this direct effect is outweighed by the influence on the air path and thus the temperature of the aspirated air.

Figure 2.48 displays the results for engine B. Test cycle 6, see Sec. 4.3, was measured once without EGR and a second time with EGR. The

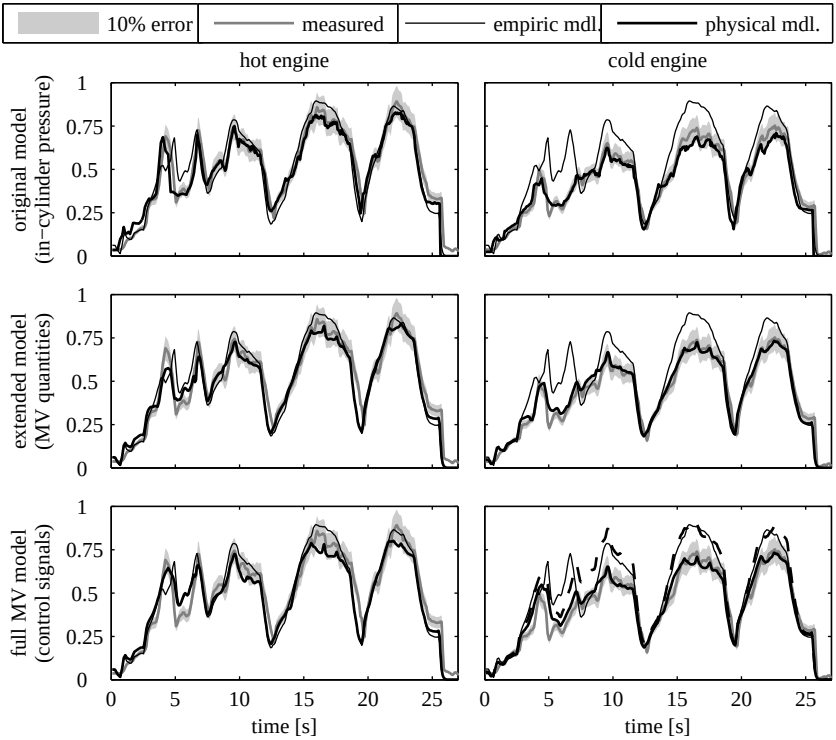

Figure 2.47: Measured NO_x mass-flow and predictions by the three model stages and the empirical model, transient engine operation, engine A. For reasons of confidentiality, normalised NO_x emissions are shown. The rows represent the three model stages. In the last row, the dashed line represents the model output obtained when the MVEM is used without the thermal dynamics of the intake and exhaust manifolds.

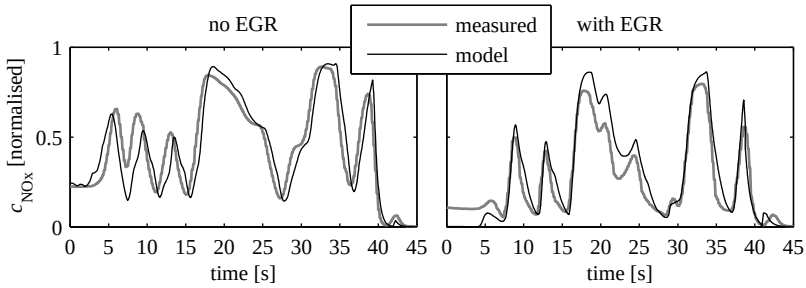

Figure 2.48: Measured NO_x emissions and predictions by the full model, engine B. The normalised NO_x concentration in the exhaust gas is shown. Only a standard exhaust-gas analyser was available, thus a variable transport delay and a first-order dynamic are visible in the measurement data. To make the model prediction more comparable to the measurement data, a first-order element with a time constant of 0.5 s was applied to the model signals. The large discrepancy in the beginning for the run with EGR is negligible since the engine is idling and thus only a low exhaust-gas mass-flow and consequently low absolute NO_x emissions prevail.

model is able to reproduce both cases. Note that the special equipment available for the validation of the model on engine A was not available on engine B.

Efficiency model The accuracy of the efficiency model during transient operation is difficult to assess. The injection quantity estimated by the ECU is not reliable, see Sec. 5.3.1. The transient external fuel scale (AVL KMA 4000) is accurate, but it is connected to the injection system by an elastic fuel line, which had to be compensated for. However, after phases with no injection or during fast changes of the fuel

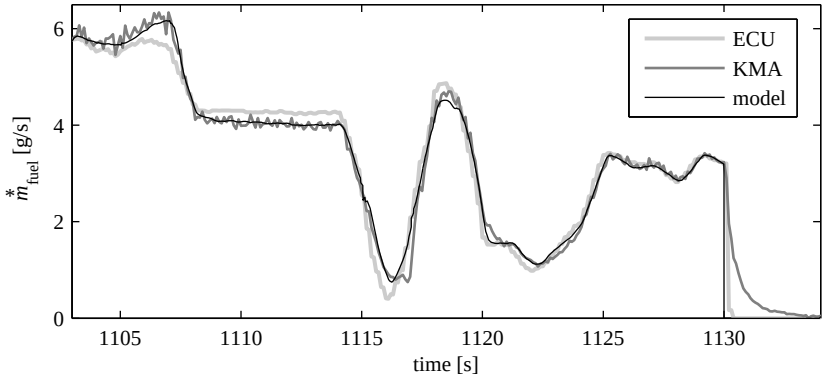

Figure 2.49: Accuracy of the torque-generation model. The inverted formulation is used in the full model for engine A. A segment of the WHTC is shown which exhibits fast changes of the load torque and thus of the fuel mass-flow. The dynamics of the elastic fuel line connecting the KMA to the common-rail system become obvious whenever the gradient of the fuel mass-flow changes rapidly.

flow, this signal still is erroneous.[14] An illustrative example is provided in Fig. 2.49.

2.7.3 Modelling recipe

The overall model can be identified according to the subdivision into a dynamic and a static part as indicated in Fig. 2.2. The adherence to this principle enables a step-wise identification since the air-path model can be used without the combustion model, e.g. for a simulation of the air path or for the development of control strategies for the air-path actuators. Accordingly, the models for the cylinder processes, the torque generation, and the pollutant emissions can be developed and identified independently from the air path. When eventually all submodels

[14] For this reason, the ECU values were used outside the region contained by the dash-dotted lines in Fig. 2.46 during the transient validation of the NO_x model.

are combined to a complete engine model, the overall accuracy has to be validated. The step-by-step procedure is outlined in the following list.

- **Measurement campaign.** Perform all the stationary and transient measurements. Only a rough engine calibration is required. At least a relatively fine engine map has to be recorded, as well as single-directed variations of all actuators included in the model need to be performed. As transient data, several fast operating-point changes are required for the identification of the thermal model, and a representative transient cycle should be recorded to validate the full model. All data is processed according to the methodology described in Appendix C.

  For an engine with EGR, two sets of measurement data should be recorded. For the first set, the EGR valve should be completely closed, or the EGR system should be physically removed. The second set represents the normal operation of the engine, including EGR.

- **Air path model.** All submodels of the air path can be identified individually, except for the turbine efficiency, which is matched to the power the compressor transfers to the air. For an engine with EGR, model versions without and with the EGR system should be derived.

- **Thermal model.** Use a forward simulation of the air-path model to manually tune the three parameters defining the thermal models for the exhaust and the intake manifolds. The effects are observed most clearly for large and fast changes of the engine power. Subsequently, check the fast dynamic response of the model using the measured transient cycle.

- **Torque generation, cylinder model, and pollutant emissions.** These models are identified individually and independently of the air-path model. After appending these models to the air-path model, the full model should be thoroughly validated on the transient cycle.

3 Numerical optimal control

In this chapter, all aspects concerning the numerical solution of optimal control problems are covered. Some of them are outlined at a high level and only the most relevant aspects are stated. For example, the topic of "Nonlinear Programming" was not analysed in detail in this work, but readily available and reliable algorithms were used. Nevertheless, an understanding of the basic ideas helps to interpret the behaviour of those algorithms and allows the parameters to be adjusted to the problem at hand. The considerations on integration errors and mesh refinement, as well as the duality within nonlinear programming are only characterised and analysed as far as required to develop the corresponding components of the optimal-control framework. In contrast, the main sections on direct collocation, the sparsity structure of the resulting mathematical program, and the calculation of derivatives are elaborated in detail.

3.1 Overview on optimal control

A continous-time optimal control problem (OCP) consists of an objective function to be minimised (or maximised) and a set of ordinary differential equations (ODE) that need to be satisfied, yielding dynamic constraints,

$$\min_{\boldsymbol{x}(\cdot),\boldsymbol{u}(\cdot)} \int_0^T L\big(\boldsymbol{x}(t),\boldsymbol{u}(t)\big)\,dt + E\big(\boldsymbol{x}(T)\big) \tag{3.1a}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}(t) - \boldsymbol{f}\big(\boldsymbol{x}(t),\boldsymbol{u}(t)\big) = 0, \quad t \in [0,T]. \tag{3.1b}$$

The integral cost $L\big(\boldsymbol{x}(t),\boldsymbol{u}(t)\big)$ is called the Lagrange term, and the end cost $E\big(\boldsymbol{x}(T)\big)$ is known as Mayer term. The combination of the two is called a Bolza objective. In the remainder of this work, only a Lagrange term is considered. Every (differentiable) Mayer term can be replaced by an equivalent Lagrange term, but the Lagrange formulation is preferable from a numerical point of view [37, Sec. 4.9].

A common extension of the unconstrained OCP (3.1) is to prescribe an initial state $\boldsymbol{x}(0) = \boldsymbol{x}_0$, or to enforce some conditions on the end state $\boldsymbol{x}(T)$. Often, "simple bounds" are imposed directly on the state variables $\boldsymbol{x}$ and on the control inputs $\boldsymbol{u}$. These limits stem from the ranges of the physical actuators represented by the control inputs, or mechanical limits on certain state variables such as rotational speed or temperature. More general "path constraints" may be imposed, which are nonlinear functions of the state variables and the control inputs.

Integral equalities or inequalities are another type of constraints that often arise in the formulation of engineering problems. These constraints may also represent a Mayer term that is rewritten in Lagrange form. Since the problem to be tackled in this thesis has to be formulated using time-variable parameters $\boldsymbol{\pi}(t)$, this special case is explicitly included in the formulation of the general OCP. Taking into account all

the extensions just outlined, the OCP reads

$$\min_{\boldsymbol{x}(\cdot),\boldsymbol{u}(\cdot)} \int_0^T L\big(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{\pi}(t)\big)\,dt \tag{3.2a}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}(t) - \boldsymbol{f}\big(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{\pi}(t)\big) = 0, \quad t \in [0,T], \tag{3.2b}$$

$$\boldsymbol{x}(0) - \boldsymbol{x}_0 = 0, \tag{3.2c}$$

$$\int_0^T \boldsymbol{g}\big(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{\pi}(t)\big)\,dt - \hat{\boldsymbol{g}} \le 0, \tag{3.2d}$$

$$\boldsymbol{c}\big(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{\pi}(t),t\big) \le 0, \quad t \in [0,T], \tag{3.2e}$$

$$\underline{\boldsymbol{x}}(t) \le \boldsymbol{x}(t) \le \overline{\boldsymbol{x}}(t), \quad t \in [0,T], \tag{3.2f}$$

$$\underline{\boldsymbol{u}}(t) \le \boldsymbol{u}(t) \le \overline{\boldsymbol{u}}(t), \quad t \in [0,T]. \tag{3.2g}$$

The nonlinear constraints on the end state are included in the path constraints (3.2e). The prescription of an initial state is kept as an explicit part of the problem formulation since this constraint is crucial for the problem at hand. The special cases of a free end time T or the presence of free (but constant) parameters are not considered.

The symbols defining the dimensions of the OCP (3.2) are introduced next. The system has n_x state variables, i.e. $\boldsymbol{x}$, $\boldsymbol{f}$, $\boldsymbol{x}_0$, $\underline{\boldsymbol{x}}$, $\overline{\boldsymbol{x}} \in \mathbb{R}^{n_\mathrm{x}}$, and there are n_u control inputs to the system, $\boldsymbol{u}$, $\underline{\boldsymbol{u}}$, $\overline{\boldsymbol{u}} \in \mathbb{R}^{n_\mathrm{u}}$. Finally, there are n_g integral constraints and n_c path constraints, i.e. $\boldsymbol{g}$, $\hat{\boldsymbol{g}} \in \mathbb{R}^{n_\mathrm{g}}$ and $\boldsymbol{c} \in \mathbb{R}^{n_\mathrm{c}}$, respectively.

3.1.1 Approaches for numerical optimal control

The most common approaches to solve continuous-time OCPs are outlined in Fig. 3.1. The top level is inspired by [64]. Other partial overviews are provided in [36, 159, 54], while [161] presents a brief historical outline. Solving the Hamilton-Jacobi-Bellman (HJB) equation, which is a partial differential equation subject to the model equations, is prac-

ticable only for systems with few state variables and control inputs. Analogously to its discrete-time equivalent, dynamic programming, it suffers the curse of dimensionality for larger systems. Similar arguments exclude the indirect approach to be applied to large and complex problems. If the first-order optimality conditions can be derived analytically, a two-point boundary-value problem (BVP) still needs to be solved. The dynamics of the co-state variables, however, are ill-conditioned, rendering a good initial guess for the general case impossible to derive.

Direct methods first discretise the full OCP, which transforms the problem to a finite-dimensional, constrained nonlinear optimisation problem. This type of problems is termed nonlinear program (NLP). Due to the sparse structure of this NLP, as well as the performance and the robustness of today's NLP solvers, this approach is a valuable means to an efficient numerical solution of large-scale and complex OCPs.

Within direct methods, two main approaches exist. The first one consists of the discretisation of the control inputs only and a forward simulation to evaluate the model. This *sequential* approach, also called *single shooting*, is often used since it is simple to implement if a suitable ODE solver is available. However, there are some important caveats. First, the resulting NLP is highly nonlinear and sensitive. A change in any control input at the beginning of the time horizon changes the solution on the entire interval, and the influence may be amplified as time progresses. This "tail wagging the dog" effect [37, Sec. 3.3-3.4] limits the length of the time horizon tractable by single shooting. Second, not only accurate, but also consistent derivatives have to be available for the solution of the NLP [37, Sec. 3.8]. An adaptive ODE solver may be accurate, but using external finite differences does not lead to consistent derivatives. Rather, ODE solvers which provide a consistent sensitivity calculation have to be used, e.g. the SUNDIALS suite of ODE solvers [103, 6]. Finally, any instabilities of the model can

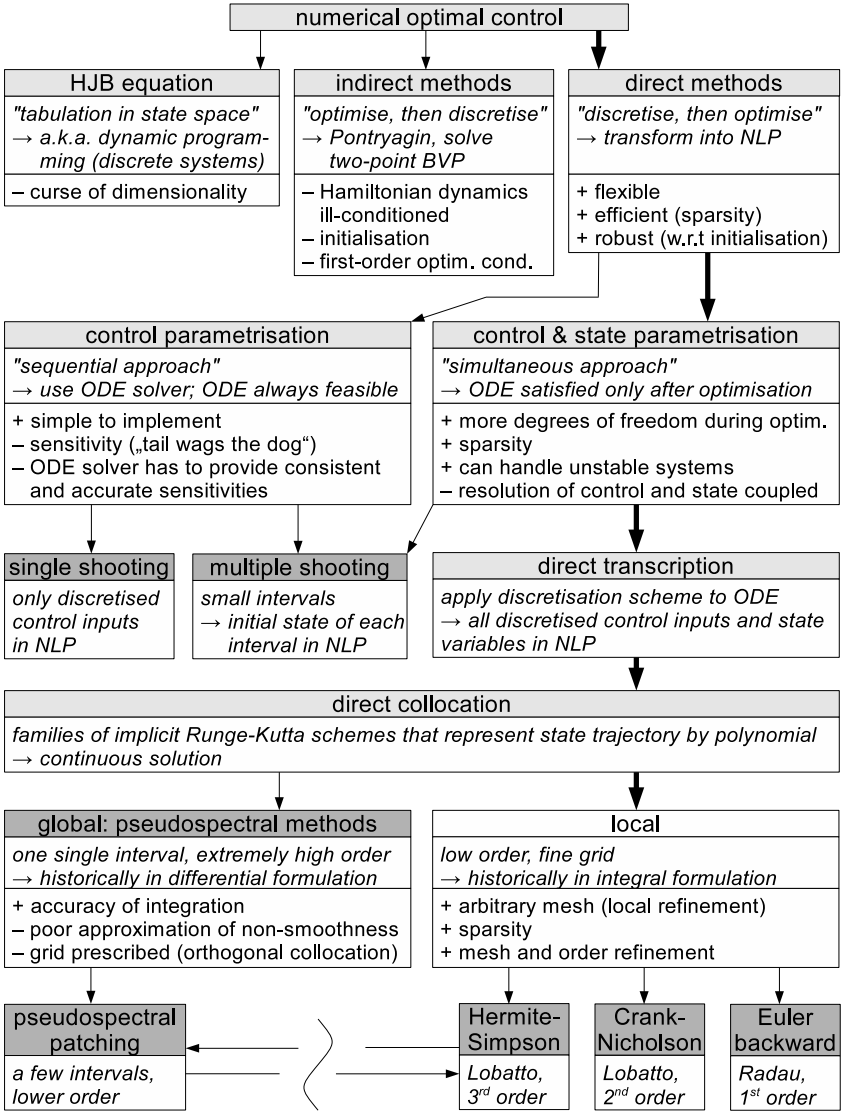

Figure 3.1: Overview of the most prominent methods for the numerical solution of optimal control problems.

lead to a failure of the ODE solver and thus a premature termination of the optimisation.

The alternative is to discretise the state trajectories along with the control inputs. The entire continous-time problem is "transcribed" into a large NLP, which fully includes the discretised state trajectories. No forward simulation is used at all, and the model equations are only satisfied after the solution of the NLP. This *simultaneous* optimisation and solution of the ODEs resolves the problems encountered by the sequential approach. Since the NLP solver can also specify the state variables, instabilites are no longer a problem. The constraints enforcing the model equations only have to be satisfied after the optimisation has converged. This additional degree of freedom during the optimisation promotes a fast convergence. Furthermore, since each of these constraints represents a single integration step, they are less nonlinear than the constraints of the sequential approach. Since an NLP solver usually considers a linearisation of the constraints at each iteration, the constraints of the simultaneous approach are thus approximated more accurately by the subproblems constructed by the solver. Finally, the NLP has a very sparse structure, which can be exploited to achieve a computationally efficient solution of the problem. One drawback, however, is the fact that the accuracy of the solution of the ODEs is coupled to the discretisation of the control inputs. An iterative mesh refinement may be necessary to obtain a sufficiently accurate ODE solution, but a fine discretisation of the control inputs may lead to numerical problems [114]. A recent overview of simultaneous approaches is provided in [40], while [26] provides insights into the origins of these methods.

Direct multiple shooting [42] fills the space between the purely sequential and the fully simultaneous approaches. The time horizon is split into small intervals, on which the control inputs are usually assumed to be constant (or piecewise linear). The initial state of each of

those intervals is included as a free variable in the NLP, and an ODE solver is used to integrate the model equations over each interval. Constraints are added to the NLP to ensure that the final state of an interval matches the prescribed initial state of the next interval. These constraints are more nonlinear than those resulting from direct transcription, but they are less restrictive than those encountered within single shooting. Direct multiple shooting shares with direct single shooting the need for custom ODE solvers that provide a consistent sensitivity calculation in a computationally efficient way.

In the context of direct transcription, collocation methods are a widely used family of integration schemes. In contrast to other Runge-Kutta schemes, they represent the state trajectories by a polynomial on each collocation interval, which provides a continuous solution. More details on direct collocation, as well as on the selection of a specific method, are provided in Sec. 3.3.

3.2 Nonlinear programming

The outline presented here summarises some sections of [143]. The goal is to provide an illustrative description of the most important concepts. Therefore, neither completeness is claimed nor a proper mathematical foundation is followed. Two main approaches exist to solve the unconstrained optimisation problem

$$\min_{\boldsymbol{\omega}} F(\boldsymbol{\omega}). \tag{3.3}$$

Both are iterative procedures which start at an initial guess $\boldsymbol{\omega}_0$ supplied by the user.

The *line search* approach first defines a direction and then searches along this direction until an acceptable step length is found. The most commonly used search direction is the *Newton direction*, which is based on a quadratic approximation of F at the current point. Not only the convergence rate is superior to the *steepest descent* direction, but the iterations become insensitive towards the scaling of the problem. Close to the solution, usually the full Newton step, i.e. the solution of the quadratic problem, is the best choice for the step length, leading to the quadratic *local convergence* rate of the Newton line-search method. At remote points, progress of the optimisation has to be ensured by considering shorter steps. It can be proven that progress is always possible along the Newton direction, and several algorithms for the search for a suitable step length exist.

In contrast, the *trust region* approach defines a region around the current point where the model for F – which again usually is a quadratic approximation – is a sufficiently accurate representation. Typically, the trust region is a ball, and the step is found by approximately solving the constrained subproblem to minimise a quadratic function inside that ball. The execution of a step is followed by an assessment of whether

the (quadratic) model is a good approximation of F in the current region. The trust-region radius to calculate the next step is then adjusted accordingly.

In proximity of the solution, both approaches reduce to a standard Newton iteration on the KKT conditions. The quadratic model becomes a close approximation of the underlying nonlinear problem and thus, always the full Newton step is accepted. Either a sufficient decrease is observed at the first trial of the line search, which usually is the Newton step itself, or the trust region is larger than the length of the Newton step. From this point of view, the line search and the trust region approaches both are *globalisation strategies* that enable the Newton iteration to converge to a minimum from remote initial points.

When the Hessian, i.e. the matrix of the second partial derivatives, is not positive definite, its inverse may not exist. In this case, the Newton direction also is not defined. In a line-search framework, the Hessian has to be made positive definite artificially, which can be combined with a matrix factorisation that is necessary for the solution anyway. Another approach is to replace the exact Hessian by a quasi-Newton approximation which is positive definite by construction, see Sec. 3.2.2. However, only a superlinear local convergence rate is retained by those methods. Due to the restriction of the search to the trust region, the trust-region subproblem is guaranteed to have a solution even when the Hessian is not positive definite. The trust-region approach thus is able to always exploit exact second-order information.

At the solution $\boldsymbol{\omega}^*$ of (3.3), the first-order necessary conditions

$$\nabla F(\boldsymbol{\omega}^*) = 0 \tag{3.4}$$

have to hold. However, a point that satisfies (3.4) could also be a saddle point instead of an extremum. To avoid this situation, the second-order sufficient conditions require the Hessian $\nabla^2 F(\boldsymbol{\omega}^*)$ to be positive definite.

Note that both conditions only identify a *local* solution, which coincides with the *global* solution only if F is *convex*. Furthermore, they only are applicable if F is smooth.

An optimisation problem that is constrained by equality and inequality constraints is called a nonlinear program (NLP). Formally, such a problem is written as

$$\min_{\boldsymbol{\omega}} \quad F(\boldsymbol{\omega}) \tag{3.5a}$$

$$\text{s.t.} \quad \boldsymbol{g}(\boldsymbol{\omega}) = 0, \quad \boldsymbol{g} \in \mathbb{R}^{n_\text{g}}, \tag{3.5b}$$

$$\boldsymbol{h}(\boldsymbol{\omega}) \leq 0, \quad \boldsymbol{h} \in \mathbb{R}^{n_\text{h}}. \tag{3.5c}$$

The "Lagrangian function" is defined as

$$\mathscr{L}(\boldsymbol{\omega}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := F(\boldsymbol{\omega}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{\omega}) + \boldsymbol{\mu}^T \boldsymbol{h}(\boldsymbol{\omega}). \tag{3.6}$$

The Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are alternatively termed the *dual variables*. Using the Lagrangian, the first-order necessary conditions for the NLP (3.5), also known as the Karush-Kuhn-Tucker (KKT) conditions, read

$$\nabla_{\boldsymbol{\omega}} \mathscr{L}(\boldsymbol{\omega}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = 0, \tag{3.7a}$$

$$\nabla_{\boldsymbol{\lambda}} \mathscr{L}(\boldsymbol{\omega}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \boldsymbol{g}(\boldsymbol{\omega}^*) = 0, \tag{3.7b}$$

$$\nabla_{\boldsymbol{\mu}} \mathscr{L}(\boldsymbol{\omega}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \boldsymbol{h}(\boldsymbol{\omega}^*) \leq 0, \tag{3.7c}$$

$$\boldsymbol{\mu}^* \geq 0, \tag{3.7d}$$

$$\mu_i^* \cdot h_i(\boldsymbol{\omega}^*) = 0, \quad i = 1, \ldots, n_\text{h}. \tag{3.7e}$$

The matrix containing the first partial derivatives of all constraints is called the Jacobian. The second-order sufficient conditions require the Hessian of the Lagrangian, $\nabla^2_{\boldsymbol{\omega}} \mathscr{L}(\boldsymbol{\omega}, \boldsymbol{\lambda}, \boldsymbol{\mu})$, projected on the null-space

of the Jacobian, to be positive definite. This projection is commonly termed the "reduced Hessian".

Equality-constrained problems can be solved by applying Newton's method to the nonlinear KKT conditions. These conditions are linearised around the current point, and the solution of the resulting system of linear equations yields the next (closer) guess to the optimal point. After some rearrangements, the so-called "KKT system" is obtained,

$$\begin{pmatrix} \nabla F(\boldsymbol{\omega}_k) \\ \boldsymbol{g}(\boldsymbol{\omega}_k) \end{pmatrix} + \begin{bmatrix} \nabla^2_{\boldsymbol{\omega}} \mathscr{L}(\boldsymbol{\omega}_k, \boldsymbol{\lambda}_k) & \nabla \boldsymbol{g}(\boldsymbol{\omega}_k) \\ \nabla \boldsymbol{g}(\boldsymbol{\omega}_k)^T & 0 \end{bmatrix} \cdot \begin{pmatrix} \boldsymbol{\omega} - \boldsymbol{\omega}_k \\ \boldsymbol{\lambda} \end{pmatrix} = 0. \tag{3.8}$$

The solution of the quadratic programming problem (QP)

$$\min_{\boldsymbol{p}} \ \frac{1}{2} \boldsymbol{p}^T \nabla^2_{\boldsymbol{\omega}} \mathscr{L}(\boldsymbol{\omega}_k, \boldsymbol{\lambda}_k) \boldsymbol{p} + \nabla F(\boldsymbol{\omega}_k)^T \boldsymbol{p} \tag{3.9a}$$

$$\text{s.t.} \quad \nabla \boldsymbol{g}(\boldsymbol{\omega}_k)^T \boldsymbol{p} + \boldsymbol{g}(\boldsymbol{\omega}_k) = 0, \tag{3.9b}$$

with $\boldsymbol{p} = \boldsymbol{\omega} - \boldsymbol{\omega}_k$, can be shown to be equivalent to the solution of (3.8). Applying Newton's method to the KKT conditions thus can be interpreted as *Sequential Quadratic Programming* (SQP).

To solve inequality-constrained problems, two fundamentally different approaches are used predominantly. The idea of SQP just introduced can be simply extended to the inequality-constrained case. These methods model (3.5) as an inequality-constrained QP at each iteration. The search direction $\boldsymbol{p}_k$ for a line search is the solution of this QP,

$$\min_{\boldsymbol{p}} \ \frac{1}{2}\boldsymbol{p}^T \nabla^2_{\boldsymbol{\omega}} \mathscr{L}(\boldsymbol{\omega}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k)\boldsymbol{p} + \nabla F(\boldsymbol{\omega}_k)^T \boldsymbol{p} \tag{3.10a}$$

$$\text{s.t.} \quad \nabla \boldsymbol{g}(\boldsymbol{\omega}_k)^T \boldsymbol{p} + \boldsymbol{g}(\boldsymbol{\omega}_k) = 0, \tag{3.10b}$$

$$\nabla \boldsymbol{h}(\boldsymbol{\omega}_k)^T \boldsymbol{p} + \boldsymbol{h}(\boldsymbol{\omega}_k) \leq 0. \tag{3.10c}$$

In trust-region methods, the trust-region constraints are added directly to this QP. A variant of SQP methods are *active-set* (AS) approaches. If the set of active inequality constraints at the solution was known beforehand, the solution would be found by solving an equality-constrained problem. AS methods update a "working set" of active inequality constraints in the outer NLP iterations, while only equality-constrained QPs are solved as inner iterations. The current working set is checked and updated by analysis of the solution of the QP. Another variant is sequential linear-quadratic programming (SLQP). First, a linear model of the problem is solved to approximate the active set at the current point, and subsequently, an equality-constrained QP is solved that includes the inequalities identified as active ones by the linear program as equality constraints.

Interior-point (IP) or *barrier methods* penalise the violation of the inequality constraints by a logarithmic barrier function. The equality-constrained problem

$$\min_{\boldsymbol{\omega}} \ F(\boldsymbol{\omega}) - \tau \sum_{i=1}^{n_\mathrm{h}} \log(-h_i) \tag{3.11a}$$

$$\text{s.t.} \quad \boldsymbol{g}(\boldsymbol{\omega}) = 0. \tag{3.11b}$$

is solved, while the barrier parameter τ is decreased iteratively. The solution for one value of this parameter is used to initialise the next

iteration. A variety of strategies to update τ exists. For highly nonlinear problems, the most successful ones are adaptive non-monotone strategies.

For constrained optimisation, the value of the objective function is not sufficient as a measure for the progress during the line search or for updating the trust radius. Rather, the decrease of the objective function has to be weighed against the decrease in the constraint violations. Different versions of these *merit functions* are used, and heuristics are applied to avoid the situation that the full Newton step is rejected for too many subsequent iterations (the so-called Maratos effect). Still, it is difficult to retain the convergence properties of the unconstrained case. Another approach is to use a *filter*, which stores Pareto-optimal pairs of the objective-function value and a scalar measure for the constraint violations. Only steps that improve at least one of those quantities are accepted, and the new pair replaces all entries in the filter which it dominates.

3.2.1 Interior-point versus active-set

Both approaches to handle inequality-constrained problems outlined above provide some advantages, but also suffer from specific drawbacks. The main difference between the two is the fact that within an SQP (or an AS) method, the structure of the QP approximation changes from inner (or outer) iteration to iteration, whereas for IP methods, this structure is invariable. As a consequence, an IP method can afford to derive a good factorisation of the KKT system once to ensure a fast solution of the individual QPs. Direct solvers for large but sparse linear systems are readily available. In contrast, an SQP method usually relies on a QP solver that detects the active set of the current QP approximation by itself. This solver thus has to deal with a constantly changing problem structure, which is usually handled by updating an initial factorisation.

However, after a certain number of updates, a re-factorisation becomes necessary for numerical reasons.

This difference is closely related to the two fundamental approaches to solve the KKT system. The full-matrix approach relies on a direct, symmetric indefinite factorisation of the KKT matrix. Various software packages performing such a factorisation are available, which are even able to exploit parallel computing, e.g. MA27/57/97 [2], MUMPS [4], or PARDISO [5]. On the other hand, the decomposition approach relies on an explicit or implicit representation of the null-space basis matrix.[1] However, for large, sparse problems the reduced Hessian generally is much more dense than the Hessian itself, requiring the application of dense linear algebra. Therefore, this approach is only computationally efficient when the number of degrees of freedom and thus the size of the reduced Hessian are small.

The most important points concerning the advantages and the drawbacks of both methods are stated next. They are mainly extracted from [85].

- It is difficult to implement SQP methods that can exploit exact second derivatives. The main pitfalls are the non-convexity of the QP subproblems when the exact Hessian is used and the fact that SQP methods often rely on custom-tailored linear algebra (e.g. for the factorisation updates) to be computationally efficient. The latter largely limits the flexibility of those algorithms w.r.t. a "modernisation" to exploit the latest developments in software and hardware technology.

[1] In contrast to the null-space approach, range-space methods [82, 83] become more efficient whenever the number of active constraints is small. Since most NLPs resulting from the transcription of OCPs include many inequality constraints that are active at the solution, null-space methods are more widely used in that context.

- In contrast, IP methods are most efficient when implemented with exact second derivatives. Furthermore, they usually converge in fewer inner iterations, even for very large problems. Since the problem structure is fixed, the inner iterations are also performed faster. The solution process relies on "off-the-shelf" linear-algebra software and is thus flexible to always exploit recent advancements, e.g. multicore processors.

- The application of IP solvers to the QPs in an SQP framework has had limited success because they are hard to warm-start [37, Sec. 4.13]. A Newton step can be interpreted as a sum of two steps. First, a normal step is taken to recover feasibility w.r.t. the linearised constraints. Second, a tangential step along the constraint surface is performed to minimise the quadratic model. An IP solver, which usually follows a solution path according to the tangential step, faces sensitivity problems when forced to step onto the path from a neighboring point.

3.2.2 Quasi-Newton approximations

The material presented in this section is extracted from [143, Chs. 6 & 18]. In Newton-type NLP solvers, the Hessian of the Lagrangian can be replaced by a quasi-Newton (QN) approximation. This substitution is possible since the current Lagrange multipliers only occur implicitly in the Hessian in the KKT system (3.8). The basic idea is to utilise the curvature information obtained along the NLP iterations to construct an increasingly accurate approximation of the exact Hessian. It can be shown that it is sufficient to have an accurate approximation of the Hessian only along the search direction.

Any QN approximation operates with the differences between two consecutive iterates and the corresponding gradients of the Lagrangian.

For constrained problems,

$$\boldsymbol{s}^T := \hat{\boldsymbol{\omega}} - \boldsymbol{\omega}, \tag{3.12}$$

$$\boldsymbol{y}^T := \nabla_\omega \mathscr{L}(\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\lambda}}) - \nabla_\omega \mathscr{L}(\boldsymbol{\omega}, \hat{\boldsymbol{\lambda}}). \tag{3.13}$$

Here, the hat indicates the values at the current iteration, i.e. the new data. In the following formulas, $\boldsymbol{B}$ denotes the approximation of the Hessian of the Lagrangian. Practical implementations are the damped BFGS update

$$\theta = \begin{cases} 1, & \text{if } \boldsymbol{s}^T\boldsymbol{y} \geq 0.2\boldsymbol{s}^T\boldsymbol{B}\boldsymbol{s} \\ (0.8\boldsymbol{s}^T\boldsymbol{B}\boldsymbol{s})/(\boldsymbol{s}^T\boldsymbol{B}\boldsymbol{s} - \boldsymbol{s}^T\boldsymbol{y}), & \text{otherwise,} \end{cases} \tag{3.14a}$$

$$r = \theta\boldsymbol{y} + (1-\theta)\boldsymbol{B}\boldsymbol{s}, \tag{3.14b}$$

$$\hat{\boldsymbol{B}} = \boldsymbol{B} - \frac{\boldsymbol{B}\boldsymbol{s}\boldsymbol{s}^T\boldsymbol{B}}{\boldsymbol{s}^T\boldsymbol{B}\boldsymbol{s}} + \frac{\boldsymbol{r}\boldsymbol{r}^T}{\boldsymbol{s}^T\boldsymbol{r}}. \tag{3.14c}$$

The damping, i.e. chosing a non-zero $\theta \in [0,1]$ if required, ensures that positive definiteness is also maintained in the constrained case. This property is desirable especially in the context of line-search methods as mentioned above.

Another widely used update is the safeguarded SR1 update

$$\hat{\boldsymbol{B}} = \begin{cases} \boldsymbol{B} + \frac{(\boldsymbol{y}-\boldsymbol{B}\boldsymbol{s})(\boldsymbol{y}-\boldsymbol{B}\boldsymbol{s})^T}{(\boldsymbol{y}-\boldsymbol{B}\boldsymbol{s})^T\boldsymbol{s}}, & \text{if } |\boldsymbol{s}^T(\boldsymbol{y}-\boldsymbol{B}\boldsymbol{s})| \geq k\|\boldsymbol{s}\|\,\|\boldsymbol{y}-\boldsymbol{B}\boldsymbol{s}\| \\ \boldsymbol{B}, & \text{otherwise.} \end{cases} \tag{3.15}$$

It is able to approximate an indefinite Hessian, which generally is the case in constrained optimisation. Trust-region methods are able to exploit this more accurate approximation. The parameter $k \approx 10^{-8}$ assures that the update is skipped if the secant equation cannot be satisfied. This infrequent skipping of the update is reasonable since in

those cases, the curvature of the Hessian in the direction of the current step is already accurately reproduced.

For large problems, QN updates exhibit several drawbacks.

1. *Loss of sparsity.* The QN approximations generally do not preserve the sparsity pattern of the exact Hessian. Enforcing the correct sparsity pattern during the update results in schemes with a poor performance. Algorithms that exploit the sparsity of the Hessian, however, have been found to exhibit superior computational performance for large-scale problems [37, Sec. 4.13].

2. *Storage vs. accuracy.* Due to the loss of sparsity, all entries of the approximated Hessian need to be stored, which limits the maximum number of NLP variables that can be handled. One alternative is to store only the last few pairs of steps and gradients and use matrix-vector products directly. These "limited-memory" updates perform well in practice, but provide less accuracy than their full-memory equivalents.

3. *Dimensionality vs. sampling.* When a high-dimensional space is sampled, perturbations in as many directions as there are dimensions would have to be performed to obtain an accurate local approximation. However, due to the memory limitations mentioned, only relatively few samples can be used.

To mitigate these limitations, the QN updates can be applied to independent small blocks of the Hessian, if applicable. Partial separability, a concept introduced by [86], describes a structural property of a nonlinear optimisation problem that allows for such a partitioned QN update of the Hessian of the Lagrangian (or of the objective, in the unconstrained case). For unconstrained optimisation, this approach was proposed and its convergence properties were analysed in [87].

Superlinear local convergence was proven, but a performance close to that obtained with exact Newton methods, which exhibit quadratic local convergence, was observed in practical experiments [129, 142]. It will be shown in Sec. 3.4.3 that partitioned updates are applicable for the NLPs resulting from direct collocation of optimal control problems.

3.2.3 NLP solvers used in this work

In the following list, the NLP solvers used during this work are characterised.

SNOPT 7.2 [84]. The "sparse nonlinear optimizer" is a proprietary solver implementing an SQP algorithm. It follows a decomposition approach by constructing and updating an LU factorisation. Its line-search method uses a merit function and is custom-tailored to work with a limited-memory BFGS update. It cannot use exact second derivatives.

IPOPT 3.11.0 [177]. The "interior point optimizer" is an open-source solver implementing a primal-dual IP method. It supports a variety of third-party linear solvers, of which MUMPS [4] with METIS preordering [3] is used here. The line-search globalisation implemented uses a filter. A limited-memory BFGS update is available, or the user may supply the exact Hessian. A step-by-step installation guide for Windows operating systems, as well as a description of the MATLAB interface are available at http://www.idsc.ethz.ch/Downloads.

WORHP 1.2-2533 [48]. The acronym of this proprietary solver (which however is free for academic use) claims that "we optimize really huge problems". The QPs within an SQP framework are solved using an IP solver that can be efficiently warm-started. The line-search method provides several merit functions or a filter. Various fully partitioned or partially overlapping BFGS updates are implemented. Depending on the operating system, different linear-algebra solvers are supported. Here, the default solver of the current Windows-compatible build, MA97, is used.

KNITRO 8.0 [50]. This proprietary solver provides three algorithms, namely an SLQP trust-region method, and two IP methods. The latter rely either on a direct solver and a line search, or on a projected conjugate-gradient method to solve trust-region subproblems. Besides the possibility of utilising exact second derivatives, the Hessian can be approximated by a full or limited-memory BFGS or the SR1 update.

3.3 Direct collocation

The concept of direct collocation to solve optimal control problems has been widely used and analysed [39, 171, 99, 32, 115, 76]. Furthermore, several software packages implement this approach, e.g. GPOPS-II (proprietary, MATLAB, http://www.gpops.org) or PSOPT (open-source, C++, http://www.psopt.org) [30]. In the introduction of the manual to PSOPT [29], a thorough overview of collocation methods is provided, and further proprietary software packages are listed.

Three groups of collocation schemes exist. Gauss collocation only contains collocation points inside the collocation interval. In contrast, Radau schemes include one boundary, whereas Lobatto schemes include both boundaries as collocation points. All three can be cast in the same framework [76]. Since the problem to be tackled is stiff, a scheme that provides stiff accuracy, i.e. that includes the right boundary, is required [95]. This requirement excludes Gauss collocation. From the remaining two, Radau collocation is preferable over Lobatto collocation for two reasons.

- Radau schemes provide stiff decay (or L-stability) [95, 49, Sec. 3.5]. L-stable schemes sufficiently damp also the very stiff components and thus generally yield smoother solutions than integration schemes that only provide A-stability, i.e. all implicit methods. A more intuitive description is provided in [16, Sec. 3.5] on page 59:

  "The practical advantage of methods with stiff decay lies in their ability to skip fine-level (i.e., rapidly varying) solution details and still maintain a decent description of the solution on a coarse level in the very stiff (not the highly oscillatory!) case."

- The integral and the differential formulations of Radau collocation are equivalent [76]. This property is necessary to construct a consistent transcription of certain optimal control problems.

Furthermore, Radau schemes exhibit algebraic stability [49, Sec. 3.5]. However, this property and the associated advantages to address differential-algebraic equations are less relevant in the context of direct collocation [69, 38].

The collocation order can be chosen arbitrarily. Historically, two different branches evolved, which is indicated in Fig. 3.1. Typically, low-order schemes are used in their integral formulation (see Sec. 3.3.2), since they are primarily applied in time-stepping methods, i.e. for the forward simulation of ODEs. Small step sizes are applied to compensate for the low order, and usually only the step size is adapted to achieve a specified accuracy. Therefore, these methods are also called "h" approaches.

On the other hand, high-order collocation in differential form is used in *pseudospectral methods*, which originally evolved in the context of partial differential equations within fluid dynamics. In their purest form, these methods do not subdivide the time horizon into multiple intervals, but represent the state variables on the full horizon as single high-order polynomials. Only the order of the approximation is increased to achieve a higher accuracy, and thus this method is also called the "p" approach. In the context of direct methods for optimal control, the terms "pseudospectral methods" and "collocation" are often used as synonyms, which indicates their common base. In fact, any order of collocation can be formulated in differential or in integral form.

Evolving from the original pseudospectral approach, *pseudospectral patching* [71] divides the time horizon in a handful of intervals. The order of the collocation is slightly reduced, and continuity conditions are enforced at the interval boundaries. The basic idea is to subdivide the problem at points where nonsmooth events occur. Nowadays, fully adaptive "hp" approaches are the state of the art [55, 56]. Depending on the curvature of the state trajectories over an interval, it is decided

whether an increase of the order (p) or a subdivision of the interval (h) is most suitable to refine the approximation. If the curvature is not uniform, i.e. the trajectory contains a sharp bend at some point, the interval is subdivided at that point. Otherwise, the order is increased.

3.3.1 The collocation equation

Consider the scalar ordinary differential equation (ODE) $\dot{x} = f(x)$ on the interval $[t_0, t_f]$. Radau collocation represents the state $x(t)$ as a polynomial, say of degree N. The time derivative of this polynomial is then equated to the values of f at N collocation points $t_0 < \tau_1 < \tau_2 \ldots < \tau_N = t_f$. The left boundary $\tau_0 = t_0$ is a non-collocated point. The notation $x_j := x(\tau_j)$ is adopted.

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_N \end{pmatrix} \approx \underbrace{[\boldsymbol{d}_0, \tilde{\boldsymbol{D}}]}_{=:\boldsymbol{D}} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_N) \end{pmatrix}. \tag{3.16}$$

Each of these N equations (index i) is a sum of linear terms and one nonlinear term,

$$d_{0,i} \cdot x_0 + \sum_{j=1}^{N} \tilde{D}_{ij} \cdot x_j - f(x_i) = 0. \tag{3.17}$$

Collocation points for the Radau (and the Lobatto) scheme(s) cannot be calculated by an explicit function. They are defined as roots of the recursively defined Legendre polynomials and have to be computed numerically [51, Sec. 2.3]. The nodes are usually calculated for the normalised interval [-1, 1] and scaled to the actual intervals as required. Chebyshev nodes may be used as initial guess, as in the calculation

method proposed on MATLAB Central (http://www.mathworks.com/matlabcentral/fileexchange/4850, and .../4775).

Lagrange interpolation by barycentric weights is used to calculate $\boldsymbol{D}$ along with the vector of the quadrature weights $\boldsymbol{w}$ [35]. Here, the step length $h = t_\mathrm{f} - t_0$ is assumed to be included in the differentiation matrix and the quadrature weights. Similar to the collocation nodes, the original differentiation matrix $\hat{\boldsymbol{D}}$ and quadrature weights $\hat{\boldsymbol{w}}$ are usually calculated for the normalised interval $[-1, 1]$, i.e. $\boldsymbol{D} := \frac{2}{h} \cdot \hat{\boldsymbol{D}}$ and $\boldsymbol{w} := \frac{h}{2} \cdot \hat{\boldsymbol{w}}$. The quadrature weights may be used to approximate the definite integral of a function $g(t)$ as $\int_{t_0}^{t_\mathrm{f}} g(t)\, dt \approx \sum_{j=1}^{N} w_j g(\tau_j)$.

The extension to a system of ODEs is considered next. Throughout the remainder of this section, the state variables are assumed to be stacked in a row vector to simplify the notation. We thus consider the system of ODEs $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ with $\boldsymbol{x}, \boldsymbol{f} \in \mathbb{R}^{1 \times n_\mathrm{x}}$. Equation (3.16) then becomes a matrix equation in $\mathbb{R}^{N \times n_\mathrm{x}}$

$$\boldsymbol{D} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{X} \end{bmatrix} = \boldsymbol{F}(\boldsymbol{X}), \tag{3.18}$$

where the rows of $\boldsymbol{X}$ and $\boldsymbol{F}$ correspond to one collocation point each. In turn, the columns of $\boldsymbol{X}$ and $\boldsymbol{F}$ represent one state variable and its corresponding right-hand side (RHS) at all collocation points. In the following, consider the notation in (3.18) as shorthand for stacking the transpose of the rows in one large column vector.

3.3.2 Differential and integral formulation

In order to discriminate the "collocation vectors" from the vectors of state variables or the RHS equations of a system of ODEs, they are designated by $\vec{\boldsymbol{x}} := (x_1, x_2, \ldots, x_N)^T$ and $\vec{\boldsymbol{f}} := \big(f(x_1), f(x_2), \ldots, f(x_N)\big)^T$. Using

this notation, the equation for the scalar case (3.16) can be rewritten as

$$\tilde{\boldsymbol{D}} \cdot \vec{\boldsymbol{x}} + \boldsymbol{d}_0 \cdot x_0 = \vec{\boldsymbol{f}} \tag{3.19a}$$

$$\Leftrightarrow \quad \vec{\boldsymbol{x}} + \tilde{\boldsymbol{D}}^{-1} \cdot \boldsymbol{d}_0 \cdot x_0 = \tilde{\boldsymbol{D}}^{-1} \cdot \vec{\boldsymbol{f}}. \tag{3.19b}$$

Thus, if the condition $\tilde{\boldsymbol{D}}^{-1} \cdot \boldsymbol{d}_0 = -1$ holds, the integral formulation of the collocation

$$\vec{\boldsymbol{x}} = x_0 + \boldsymbol{A} \cdot \vec{\boldsymbol{f}} \tag{3.20}$$

is obtained, where $\boldsymbol{A} := \tilde{\boldsymbol{D}}^{-1}$. As mentioned above, the differential and the integral formulations of Radau collocation are equivalent. Therefore, the matrix of the Runge-Kutta coefficients can be calculated as $\boldsymbol{A} := \tilde{\boldsymbol{D}}^{-1}$.[2]

For Lobatto collocation, the differentiation matrix $\boldsymbol{D}$ has an additional "zeroth row" since the left boundary is included as a collocation point. The fact that this differentiation matrix is not invertible is obvious since in the integral formulation, the equation for the first point – the left boundary itself – is trivially satisfied when the first row of $\boldsymbol{A}$ is chosen to contain only zeros. A matrix with a zero row, however, is singular. The Runge-Kutta coefficients $\boldsymbol{A}$ thus have to be calculated numerically, and the integral formulation has one equation less per interval than the differential formulation. Therefore, the two formulations actually constitute different problems.

Exemplary comparison Although the differential (3.16) and the integral (3.20) formulations of the Radau collocation are equivalent, different constraints for the NLP result. In order to analyse the two formula-

[2] $\boldsymbol{A}$ is the upper-right partition of the Butcher tableau [49, Sec. 2.3]. The nodes are known and the lower part of the table, the integration weights, are obviously equal to the last row of $\boldsymbol{A}$ for stiffly accurate collocation schemes.

tions, the simple ODE

$$\dot{x} = -ax + k := f(x), \quad x(0) = x_0 \tag{3.21}$$

with the explicit solution

$$x = \frac{k}{a} - \left(\frac{k}{a} - x_0\right) \cdot \exp(-at) \tag{3.22}$$

is considered. Its solution is approximated by Radau collocation. The state x is initialised with the value x_0 at all discretisation points, and MATLAB's Newton iteration `fsolve` is applied to iteratively approach the constraints (3.16) or (3.20), respectively.[3] For the following analysis, the problem parameters are chosen as $k = 1$, $a = 200$, and $x_0 = 10$. The ODE is solved on the interval $t \in [0, 0.1]$.

Figure 3.2 shows intermediate iterations for various discretisation schemes. The differential formulation starts to approximate the shape of the solution after few iterations already. Conversely, the integral formulation produces highly oscillating intermediate approximations. Above all, it has difficulties to achieve a smooth transition between the intervals for all non-pseudospectral approaches. For higher-order methods, this behaviour can also be observed for the differential formulation, but in a less severe form. The differential formulation thus seems to be preferable also for the low-order "h" approach.[4]

Some preliminary tests on the problem to be solved in this work were conducted. For this complex real-world problem as well, the differ-

[3] Since an autonomous ODE is considered, no discretised control inputs are present in the NLP.

[4] In theory, a Newton method is invariant w.r.t. linear transformations such as $\tilde{\boldsymbol{D}}^{-1}$. Therefore, the different steps produced by the integral and the differential formulations could be caused by the specific implementation of `fsolve`, but this conjecture was not validated.

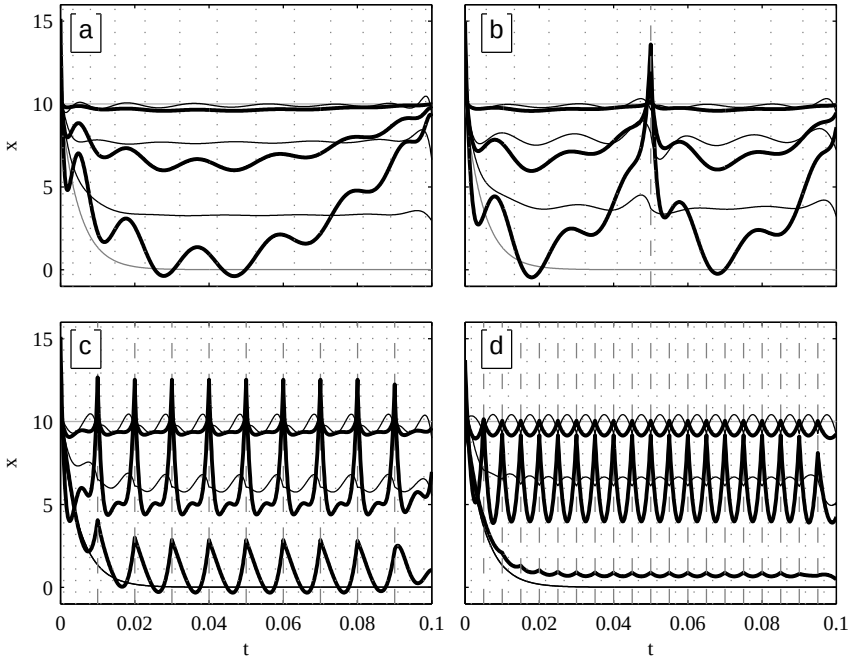

Figure 3.2: Numerical solution of the ODE (3.21) by the differential formulation (thin line) and the integral formulation (bold line) of the Radau collocation. a) Pseudospectral method (order 15), b) pseudospectral patching (2 intervals, order 8 each), c) order 4 and $h = 0.01$, d) order 2 and $h = 0.005$. Iterations 1, 3 and 4 shown in a) and b), iterations 2, 4 and 5 shown in c) and d). All versions converge after 6 iterations, except for the integral formulation of d), which requires 7 iterations.

ential formulation proves to be preferable over the integral formulation. Any NLP solver required more iterations to solve the same problem in integral than the equivalent differential formulation. For higher orders, the effect becomes even more pronounced.

In the differential formulation, the model functions are not scaled by the step size of the discretisation scheme. Therefore, only the linear part of the discretised continuity constraints are scaled by the step size, which do not occur in the second derivatives, however. A more similar numerical condition of the QP approximations for different orders and step sizes results. This fact could explain the faster convergence of the NLP solvers when the differential formulation is used, especially for higher orders.[5]

[5] Moreover, the Jacobian of the NLP resulting from the integral formulation is denser than that corresponding to the differential formulation, see Sec. 3.4 and especially Fig. 3.3. In the differential formulation, each dynamic constraint is a sum of several linear terms and a single nonlinear term. The corresponding constraints in the integral formulation are weighted sums over the nonlinear model function evaluated at all collocation nodes. Hence, the block of interval 3 would be completely dense in Fig. 3.3.

3.3.3 Application to optimal control problems

For the analysis performed here, the control-input vector is assumed to be a row vector as well, i.e. $\boldsymbol{u} \in \mathbb{R}^{1\times n_\mathrm{u}}$. A more compact representation of the original problem (3.2) is used, namely the constraint for a fixed initial state (3.2c) as well as the simple bounds on the control inputs (3.2g) and on the state variables (3.2f) are included in the more general path constraints (3.2e). Furthermore, the time-variable parameters are not explicitly included, which however does not impair the results of all considerations in this and the subsequent sections. Finally, the integral constraints are assumed to be formulated such that the integrals have to be nonpositive.[6] The more compact formulation of the OCP reads

$$\min_{\boldsymbol{x}(\cdot),\boldsymbol{u}(\cdot)} \int_0^T L\big(\boldsymbol{x}(t),\boldsymbol{u}(t)\big)\,dt \tag{3.23a}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}(t) - \boldsymbol{f}\big(\boldsymbol{x}(t),\boldsymbol{u}(t)\big) = 0, \quad t\in[0,T], \tag{3.23b}$$

$$\int_0^T \boldsymbol{g}\big(\boldsymbol{x}(t),\boldsymbol{u}(t)\big)\,dt \le 0, \tag{3.23c}$$

$$\boldsymbol{c}\big(\boldsymbol{x}(t),\boldsymbol{u}(t),t\big) \le 0, \quad t\in[0,T]. \tag{3.23d}$$

Direct transcription discretises all functions and integrals by consistently applying an integration scheme. Here, $k = 1,...,m$ integration intervals $[t_{k-1}, t_k]$ are used with $0 = t_0 < t_1 ... < t_m = T$. The number of collocation points N_k can be different for each interval. Summing up the collocation points throughout all integration intervals results in a total of $M = l(m, N_m)$ discretisation points. The "linear index" l thereby corresponds to collocation node i in interval k,

$$l := l(k,i) = i + \sum_{\alpha=1}^{k-1} N_\alpha. \tag{3.24}$$

[6] This formulation is obtained by a simple shift by $\hat{\boldsymbol{g}}$.

The following constrained optimisation problem results.

$$\min_{\boldsymbol{x}_\bullet, \boldsymbol{u}_\bullet} \quad \sum_{l=1}^{M} W_l \cdot L\left(\boldsymbol{x}_l, \boldsymbol{u}_l\right) \tag{3.25a}$$

$$\text{s.t.} \quad \boldsymbol{D}^{(k)} \begin{bmatrix} \boldsymbol{X}^{(k-1)}_{N_{k-1},\bullet} \\ \boldsymbol{X}^{(k)} \end{bmatrix} - \boldsymbol{F}\left(\boldsymbol{X}^{(k)}, \boldsymbol{U}^{(k)}\right) = 0,\ k = 1 \ldots m, \tag{3.25b}$$

$$\sum_{l=1}^{M} W_l \cdot \boldsymbol{g}(\boldsymbol{x}_l, \boldsymbol{u}_l) \leq 0, \tag{3.25c}$$

$$\boldsymbol{c}_l(\boldsymbol{x}_l, \boldsymbol{u}_l) \leq 0,\ l = 1 \ldots M. \tag{3.25d}$$

The vector of the "global" quadrature weights $\boldsymbol{W}$ results from stacking the vectors of the quadrature weights $\boldsymbol{w}^{(k)}$ of each interval k after removing the first element, which is zero. For the first interval, $\boldsymbol{X}^{(k-1)}_{N_{k-1},\bullet}$ is the initial state $\boldsymbol{x}(0)$.

The Lagrangian of the NLP (3.25) is the sum of the objective (3.25a) and all constraints (3.25b), (3.25c) and (3.25d), which are weighted by the Lagrange multipliers $\boldsymbol{\lambda}$.[7] To simplify the notation, the Lagrange multipliers are grouped according to the problem structure in (3.25). The $n_\mathrm{x} \cdot N_k$ multipliers for the discretised dynamic constraints on each integration interval k are denoted by $\boldsymbol{\lambda}^{(k)}_\mathrm{d}$, the n_g multipliers for the integral inequalities are stacked in $\boldsymbol{\lambda}_\mathrm{g}$ and the n_c multipliers for the path constraints at each discretisation point l are gathered in the vector $\boldsymbol{\lambda}_{\mathrm{c},l}$.

[7] In contrast to Sec. 3.2, no different symbols are used to discriminate the dual variables of the equality and the inequality constraints. The multipliers of all constraints are uniformly denoted by $\boldsymbol{\lambda}$.

3.4 Sparsity structure

The sparsity of a problem refers to the fact that not all functions (e.g. the objective function and all constraints) depend on all free variables. Of interest are the "orders of dependency" that actually are present within a certain representation of the problem. For example, in a dynamic model of a physical system, usually each control input and state variable eventually influences all state variables. However, when a linearisation of the system is considered, only the direct or "first-order" influences are captured in the system matrices.

In the QP subproblems within an NLP solver (3.9)[8], the gradient of the objective function, the Jacobian of the constraints, and the Hessian of the Lagrangian occur. This information thus has to be provided to the solver. Some solvers require the objective gradient to be provided as additional (first) row in the constraint Jacobian. This formulation is adopted here for a compact presentation. The sparsity patterns of the constraint Jacobian and of the Hessian also define the structure of the KKT matrix in (3.8), which is important for the performance of direct solvers for the KKT system.

In the next sections, the separability and the sparsity structure of an NLP resulting from direct collocation of an OCP are analysed. Furthermore, it is shown how this structure can be exploited to reduce the function evaluations to a minimum, to apply partitioned QN updates, and to apply parallel computing to the model evaluations.

3.4.1 Structure and separability

The objective (3.25a), the integral inequalities (3.25c) and the path constraints (3.25d) are inherently separated w.r.t. time, i.e. the single terms

[8] Since the QPs at the lowest level always are equality-constrained problems, with possibly the active set of inequality constraints included as additional equality constraints, the equality-constrained version is referred to.

are pairwise disjoint in $\boldsymbol{x}_l$ and $\boldsymbol{u}_l$. The focus thus is on the structure and the separability of the defect constraints (3.25b). For the derivation, f, x and u are assumed to be scalar. The extension to the vector-valued case is straightforward and will be provided after the results for the scalar case are obtained.

Consider the term of the Lagrangian representing the defect constraints (3.25b) for interval k,

$$\tilde{\mathscr{L}}_{\mathrm{d}}^{(k)} := \sum_{i=1}^{N_k} \lambda_{\mathrm{d},i}^{(k)} \left(d_{0,i}^{(k)} \cdot x_{N_{k-1}}^{(k-1)} + \sum_{j=1}^{N_k} \tilde{D}_{ij}^{(k)} x_j^{(k)} - f\big(x_i^{(k)}, u_i^{(k)}\big) \right). \quad (3.26)$$

This formulation constitutes a separation in the dual variables, i.e. the Lagrange multipliers. By collecting terms at each collocation point and accounting for the d_0 terms in the previous interval, a separation in the primal variables is obtained,

$$\mathscr{L}_{\mathrm{d}}^{(k)} = \sum_{i=1}^{N_k} \left(\sum_{j=1}^{N_k} \tilde{D}_{ji}^{(k)} \lambda_{\mathrm{d},j}^{(k)} x_i^{(k)} + \delta_i^{(k)} \sum_{j=1}^{N_{k+1}} d_{0,j}^{(k+1)} \lambda_{\mathrm{d},j}^{(k+1)} x_i^{(k)} - \lambda_{\mathrm{d},i}^{(k)} f\big(x_i^{(k)}, u_i^{(k)}\big) \right), \quad (3.27)$$

with

$$\delta_i^{(k)} = \begin{cases} 1, & \text{if } i = N_k \text{ and } k \neq m, \\ 0, & \text{else.} \end{cases} \quad (3.28)$$

Each term in (3.27) is the collocation-point separated part of the Lagrangian which stems from the defect constraints,

$$\mathscr{L}_{\mathrm{d},i}^{(k)} = \left(\sum_{j=1}^{N_k} \tilde{D}_{ji}^{(k)} \lambda_{\mathrm{d},j}^{(k)} + \delta_i^{(k)} \sum_{j=1}^{N_{k+1}} d_{0,j}^{(k+1)} \lambda_{\mathrm{d},j}^{(k+1)} \right) \cdot x_i^{(k)} - \lambda_{\mathrm{d},i}^{(k)} f\big(\omega_i^{(k)}\big), \quad (3.29)$$

where the notation

$$\omega := (x, u), \quad \omega_i^{(k)} := \left(x_i^{(k)}, u_i^{(k)}\right) \tag{3.30}$$

is used. Its gradient w.r.t. the primal variables is

$$\nabla_\omega \mathscr{L}_{\mathrm{d},i}^{(k)} = \left(\boldsymbol{\lambda}_{\mathrm{d}}^{(k)T} \tilde{D}_{\bullet i}^{(k)} + \delta_i^{(k)} \boldsymbol{\lambda}_{\mathrm{d}}^{(k+1),T} \boldsymbol{d}_0^{(k+1)}, 0\right) - \lambda_{\mathrm{d},i}^{(k)} \frac{\partial f(\omega_i^{(k)})}{\partial \omega}. \tag{3.31}$$

Its Hessian is simply[9]

$$\nabla_\omega^2 \mathscr{L}_{\mathrm{d},i}^{(k)} = -\lambda_{\mathrm{d},i}^{(k)} \frac{\partial^2 f(\omega_i^{(k)})}{\partial \omega^2}. \tag{3.32}$$

Vector-valued case and complete element Lagrangian For multiple control inputs and state variables, the primal variables $\boldsymbol{\omega}_l$ at each collocation point become a vector in $\mathbb{R}^{1\times(n_\mathrm{u}+n_\mathrm{x})}$. Consistently, the gradient of a scalar function w.r.t. $\boldsymbol{\omega}$ is considered a row vector. Thus, the model Jacobian $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\omega}}$ is a matrix in $\mathbb{R}^{n_\mathrm{x}\times(n_\mathrm{u}+n_\mathrm{x})}$, and the Hessian of each model-function element s, $\frac{\partial^2 \boldsymbol{f}_s}{\partial \boldsymbol{\omega}^2}$, is a square matrix of size $(n_\mathrm{u}+n_\mathrm{x})$. The multiplier $\boldsymbol{\lambda}_{\mathrm{d},i}^{(k)}$ itself also becomes a vector in $\mathbb{R}^{n_\mathrm{x}}$. All terms involving $\boldsymbol{f}$, its Jacobian, or its Hessian therefore turn into sums.

The full element Lagrangian $\mathscr{L}_i^{(k)}$ consists of the terms of the defect constraints $\mathscr{L}_{\mathrm{d},i}^{(k)}$ as derived above, plus the contributions of the

[9] As mentioned in Sec. 3.3.2, the step size (contained in $\boldsymbol{D}$) does not occur in the Hessian, which is not the case for the integral formulation. This fact leads to a generally better condition of the differential formulation for small step sizes [114].

objective, the integral inequalities and the path constraints,

$$\begin{aligned}\mathscr{L}_i^{(k)} = {} & W_l \cdot L(\boldsymbol{\omega}_l) + \\ & \boldsymbol{x}_i^{(k)} \cdot \left(\sum_{j=1}^{N_k} \tilde{D}_{ji}^{(k)} \boldsymbol{\lambda}_{\mathrm{d},j}^{(k)} + \delta_i^{(k)} \sum_{j=1}^{N_{k+1}} d_{0,j}^{(k+1)} \boldsymbol{\lambda}_{\mathrm{d},j}^{(k+1)} \right) \\ & - \boldsymbol{f}\left(\boldsymbol{\omega}_i^{(k)}\right) \boldsymbol{\lambda}_{\mathrm{d},i}^{(k)} + W_l \cdot \boldsymbol{\lambda}_{\mathrm{g}}^T \, \boldsymbol{g}(\boldsymbol{\omega}_l) + \boldsymbol{\lambda}_{\mathrm{c},l}^T \, \boldsymbol{c}(\boldsymbol{\omega}_l).\end{aligned} \tag{3.33}$$

The Lagrangian of the full NLP is obtained by summing these element Lagrangians, which are separated in the primal variables. Its Hessian thus is a perfect block-diagonal matrix with uniformly sized square blocks of size $(n_{\mathrm{u}} + n_{\mathrm{x}})$.

Extension to Gauss and Lobatto collocation Gauss collocation does not include the right interval boundary. Thus, the terms involving $\boldsymbol{d}_0$ can be included locally in each interval, which simplifies the separation in the primal variables. However, the continuity constraint

$$\boldsymbol{x}_0^{(k+1)} = \boldsymbol{x}_0^{(k)} + \boldsymbol{w}^{(k)T} \boldsymbol{F}\left(\boldsymbol{X}^{(k)}, \boldsymbol{U}^{(k)}\right) \tag{3.34}$$

has to be introduced for each interval. Similarly to the procedure above, this constraint can be separated. The quadrature weights $\boldsymbol{w}^{(k)}$ are stacked in $\boldsymbol{W}$ without any modification.

Lobatto collocation includes both boundaries as collocation points. Thus, the matrix $\boldsymbol{D}$ in (3.16) and (3.18) has an additional "zeroth" row, and the argument of $\boldsymbol{F}$ becomes $\left[\boldsymbol{x}_0^T, \boldsymbol{X}^T\right]^T$ in (3.18). The additional term

$$-\delta_i^{(k)} \lambda_{\mathrm{d},0}^{(k+1)} f\left(\omega_i^{(k)}\right) \tag{3.35}$$

arises in (3.29), which however does not impair the separability w.r.t. the primal variables. Each element of $\boldsymbol{W}$ corresponding to the interval

boundary between any two neighbouring intervals k and $k+1$ is a sum of the two weights $w_{N_k}^{(k)}$ and $w_0^{(k+1)}$.

For the first interval, any terms in $\boldsymbol{x}_0^{(1)}$ and $\boldsymbol{f}\left(\boldsymbol{x}_0^{(1)}, \boldsymbol{u}_0^{(1)}\right)$ have to be included in $\mathscr{L}_{\mathrm{d}}^{(0)}$ since there is no preceding interval. This special case has to be accounted for in the implementation of Radau or Lobatto collocation schemes.

3.4.2 Sparsity exploitation

The separability of the problem w.r.t. the primal variables allows a perfect exploitation of the problem sparsity. In fact, the Jacobian of the objective and the constraints, as well as the Hessian of the Lagrangian can be constructed from the first and second derivatives of the nonlinear model functions L, $\boldsymbol{f}$, $\boldsymbol{g}$ and $\boldsymbol{c}$ at each discretisation point, as proposed in [149].

The Jacobian entries for the objective and the integral constraints are the corresponding integrands, weighted by the quadrature weights. The first derivatives of the path constraints are just the first derivatives of the constraint functions themselves. The Jacobian entries of each discrete continuity constraint (3.17) consists of a constant part in all of the collocation points of a given collocation interval, as well as the first derivatives of the model function at the collocation point under consideration.

When constructing the Jacobian of the NLP from the first derivatives of the integrands, the model functions, and the constraint functions, the critical part is to place the elements in the correct place in the Jacobian. This indexing can either be done right after the calculation of the derivatives at each discretisation point, or only after all derivatives are available. In a sequential computation, these two approaches are equivalent. However, if parallel computing is to be applied, the second variant is much simpler to be implemented since only the derivative

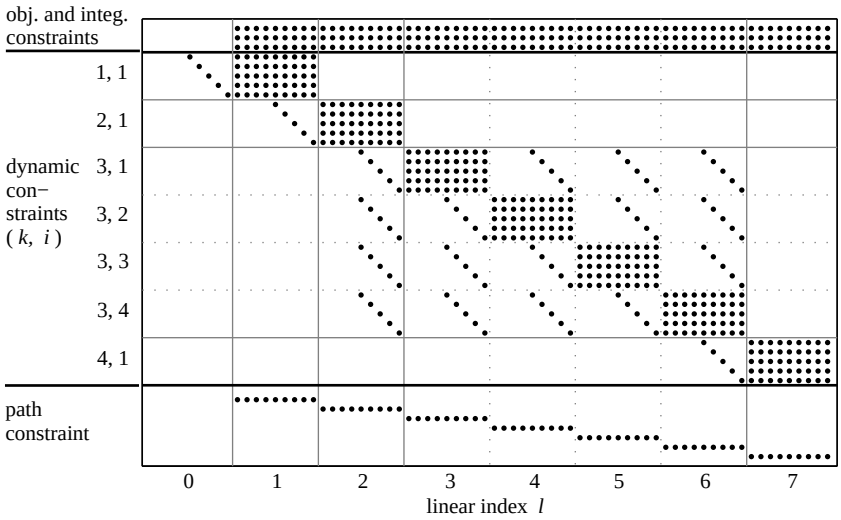

Figure 3.3: Sparsity structure of the Jacobian of the NLP resulting from transcribing a continuous-time OCP by differential Radau collocation. Four collocation intervals are used, all with collocation order 1 except for the third interval which is of order 4. Characteristically for Radau collocation, the zeroth point (left boundary of the time horizon considered) only contributes linearly to the first dynamic constraint of the first interval. The variables at each discretisation point are ordered as $\left(\boldsymbol{u}_l^T, \boldsymbol{x}_l^T\right)^T$.

calculation at each point has to be parallelised, which is a straightforward process. This issue is discussed in Sec. 3.4.4 in some more detail. The optimal-control framework described in Sec. D implements both approaches.

Figure 3.3 displays the sparsity pattern of the Jacobian of an exemplary OCP with two integral constraints and one path constraint. The underlying model has four control inputs and five state variables. For illustration purposes, only four collocation intervals are used, on which differential Radau collocation of order 1 is applied, except for the

third one, which is of order 4. The rows represent the objective and all nonlinear constraints, ordered as indicated in the plot. Each column represents a single NLP variable, i.e. one control input or state variable at one discretisation point. They are grouped into all control inputs and state variables at each point with linear index l.

Note that for the integral formulation, the constant diagonal entries inside the 4×4 block would become variable dense blocks. All coefficients describing the integral collocation conditions are multiplied with the model functions, see (3.20). Therefore, only the state variables at the left boundary and at the collocation point under consideration occur as linear contributions to the constraint.

As mentioned above, the Hessian of the Lagrangian has a block-diagonal structure. In fact, the nonlinear part of the element Lagrangian (3.33) is a weighted sum of the integrand of the objective, all elements of the model-function vector, the integrands of the integral constraints, and the path constraints. Therefore, its Hessian can be calculated as a sum of the Hessian matrices corresponding to all nonlinear functions, weighted by the same multipliers. It is therefore only necessary to calculate at each discretisation point l the Jacobian of all model functions and, if an exact Newton method is applied, the Hessian of each model function to construct the corresponding matrices of the NLP.

3.4.3 Partitioned quasi-Newton updating

The separability of the NLP can also be exploited when calculating QN approximations of the Hessian. Recall that these iterative updates collect information about the curvature of the Lagrangian by sampling its gradient at various points. Naturally, these points are chosen as the steps of a Newton-type iteration that solves the NLP.

Using approximations of the small diagonal blocks, i.e. exploiting the separability illustrated in Sec. 3.4.1, resolves the problems when QN approximations are applied to large problems, as listed in Sec. 3.2.2.

1. In contrast to a QN update of the full Hessian, the sparsity pattern of the true Hessian is exactly reproduced.
2. Only $M \cdot (n_\mathrm{x} + n_\mathrm{u})^2$ numbers have to be stored, as compared to $M^2 \cdot (n_\mathrm{x} + n_\mathrm{u})^2$ for the full Hessian.
3. Since the dimension of each diagonal block is small, a good approximation is already obtained after few iterations of the Hessian update [129, 142].

By exploiting the problem sparsity when calculating the gradient of the Lagrangian *and* also for the QN updates, the following goal is achieved: *"Obtain an accurate approximation of the Hessian of the Lagrangian, which exhibits the sparsity pattern of the exact Hessian, by only calculating the gradients of the model functions at each collocation point."*

The Hessian block $\boldsymbol{B}_l$ corresponding to the element Lagrangian (3.33) at collocation point l is approximated as follows. In the difference of the gradients, all linear terms cancel. Thus, (3.13) becomes

$$\begin{aligned}
\boldsymbol{y}_l^T &= \nabla_\omega \mathscr{L}_l(\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\lambda}}) - \nabla_\omega \mathscr{L}_l(\boldsymbol{\omega}, \hat{\boldsymbol{\lambda}}) \\
&= W_l \cdot \left(\frac{\partial L(\hat{\boldsymbol{\omega}}_l)}{\partial \boldsymbol{\omega}} - \frac{\partial L(\boldsymbol{\omega}_l)}{\partial \boldsymbol{\omega}} \right) + \\
&\quad \boldsymbol{\lambda}_{\mathrm{d},l}^T \cdot \left(\frac{\partial \boldsymbol{f}(\boldsymbol{\omega}_l)}{\partial \boldsymbol{\omega}} - \frac{\partial \mathbf{f}(\hat{\boldsymbol{\omega}}_l)}{\partial \boldsymbol{\omega}} \right) + \\
&\quad W_l \cdot \boldsymbol{\lambda}_\mathrm{g}^T \cdot \left(\frac{\partial \mathbf{g}(\hat{\boldsymbol{\omega}}_l)}{\partial \boldsymbol{\omega}} - \frac{\partial \mathbf{g}(\boldsymbol{\omega}_l)}{\partial \boldsymbol{\omega}} \right) + \\
&\quad \boldsymbol{\lambda}_{\mathrm{c},l}^T \cdot \left(\frac{\partial \mathbf{c}(\hat{\boldsymbol{\omega}}_l)}{\partial \boldsymbol{\omega}} - \frac{\partial \boldsymbol{c}(\boldsymbol{\omega}_l)}{\partial \boldsymbol{\omega}} \right).
\end{aligned} \tag{3.36}$$

Recall that the linear index l is defined such that $\boldsymbol{\lambda}_{\mathrm{d},l} = \boldsymbol{\lambda}_{\mathrm{d},i}^{(k)}$. The QN updates (3.14) or (3.15) are applied to each diagonal block $\boldsymbol{B}_l$ individually, with $\boldsymbol{s}_l^T = \hat{\boldsymbol{\omega}}_l - \boldsymbol{\omega}_l$ and $\boldsymbol{y}_l$ given by (3.36).

3.4.4 Parallelisation of the model evaluations

Parallel computing can be used to speed up the calculation of the derivative information provided to the NLP solver during each iteration. The parallelisation of the model evaluations is scheduled by the discretisation steps. The total number of steps is distributed into as many sets as there are parallel processing units. For very large problems, it becomes worthwile to also include the actual construction of the Jacobian from the model derivatives in the parallelisation. A sparse matrix representation has to be used in order to avoid facing memory limitations for large problems. MATLAB uses a compressed column format to store sparse matrices. Therefore, whenever a previously populated column is modified, several shift registers need to be updated. The indexing thus is slow and it is advisable to also compose individual partitions of the Jacobian in parallel. Only for small problems, the distribution and gathering of the data outweighs the gains of this full parallelisation.

Figure 3.4 (top) shows the absolute runtime for the calculation of one Jacobian of the NLP for the problem at hand. Either only the model derivatives are computed in parallel, or the construction of the Jacobian is parallelised as well. One, two or four shared-memory processing units are used. For all cases, the runtime is proportional to the number of time steps, as expected.

The bottom plot shows the parallelisation efficiency. In the ideal case, a speedup factor equal to the number of parallel units would result. However, communication overhead and the necessary distribution and gathering of data reduce this gain. The parallelisation efficiency is the ratio of the speedup actually achieved to the ideal speedup. As

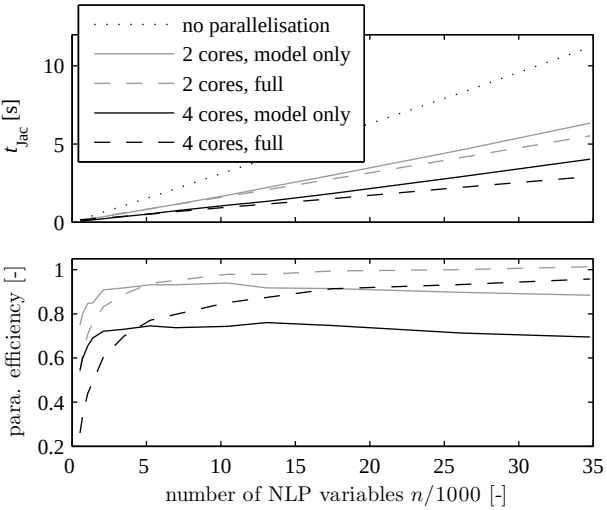

Figure 3.4: Absolute runtime and parallelisation efficiency for the Jacobian calculation. Either only the model evaluations ("model only") or also the Jacobian construction is parallelised ("full").

explained above, it is only for large problems that it becomes advantageous to parallelise the Jacobian construction along with the model evaluations. In fact, the parallelisation efficiency starts to decrease again for large problems if the Jacobian construction is not parallelised along with the model evaluations.

Figure 3.5 displays the same two plots for the calculation of the Hessian. Again, the computational time increases linearly with the size of the problem, but one evaluation takes substantially longer than in the case of the Jacobian. Therefore, the parallelisation has a more substantial impact on the overall runtime when an exact Newton method is applied. In the case of the Hessian, the indexing is negligible as compared to the model evaluations since its block-diagonal structure complies with the sparse compressed-column format. The stagnation

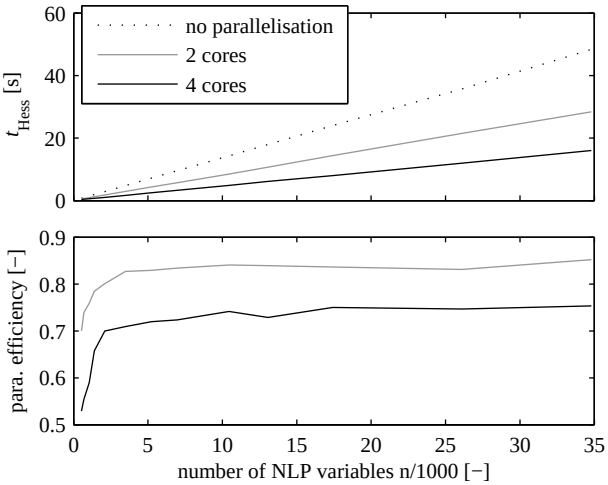

Figure 3.5: Absolute runtime and parallelisation efficiency for the Hessian calculation. The time for indexing is negligible compared to the time required for the model evaluations.

of the parallelisation efficiency for large problems is a result of the throttling of the clock rate by the system in order to prevent overheating of the CPU.

3.5 Mesh refinement

The idea of an iterative mesh refinement is to first solve a coarse approximation of the continuous-time OCP. This solution is then used to identify regions where the discretisation needs refinement. As described in Sec. 3.3, several approaches for the mesh refinement are possible. Either the step size, the collocation order, or both are adjusted. Similarly, there are various ways of estimating the current error of the ODE solution, which then implies the degree of refinement required on each discretisation interval.

Mesh refinement is not analysed in depth, but is considered only to enable a thorough analysis of the direct collocation method. The aim is to derive qualitative results concerning the impact of applying an iterative refinement of the problem. Therefore, only step-size refinement is considered, and consistently, the error is estimated by the local truncation error. A more detailed analysis of the convergence of the discretised problem towards the continuous one is provided in [115] for Radau collocation. From a theoretical point of view, the state variables, the control inputs, and the dual variables exhibit $O\left(h^{s+1}\right)$ convergence, where s is the order of the polynomials representing the state trajectories. Accordingly, there are $s+1$ collocation nodes. Although numerical experiments reveal that the state variables converge with $O\left(h^{s+2}\right)$, the theoretical result is used here.

3.5.1 Truncation error

When approximating the solution of an ODE on a given interval by a polynomial of degree s, by definition the truncation error $\boldsymbol{\tau}$ is of order $s+1$,

$$\boldsymbol{x} = \boldsymbol{x}^{(1)} + \boldsymbol{\tau}^{(1)}, \tag{3.37a}$$

$$\boldsymbol{\tau}^{(1)} = \boldsymbol{c} \cdot h^{s+1}. \tag{3.37b}$$

Here, $\boldsymbol{x}$ denotes the state at the end of the interval, i.e. $\boldsymbol{x} := \boldsymbol{x}(t_k + h)$, and $\boldsymbol{x}^{(p)}$ is its approximation using p integration steps. The factors $\boldsymbol{c}$ depend on the model function $\boldsymbol{f}$ and therefore are not constant. However, over the time horizon of one integration step, assuming a constant value of $\boldsymbol{c}$ is reasonable.[10]

By dividing the interval into p equal steps, a more accurate approximation to the exact solution is obtained.

$$\boldsymbol{x} = \boldsymbol{x}^{(p)} + \boldsymbol{\tau}^{(p)}, \tag{3.38a}$$

$$\boldsymbol{\tau}^{(p)} = p \cdot \boldsymbol{c} \cdot \left(\frac{h}{p}\right)^{s+1} = \frac{\boldsymbol{c} \cdot h^{s+1}}{p^s} = \frac{\boldsymbol{\tau}^{(1)}}{p^s}. \tag{3.38b}$$

By equating the expressions for the exact solution $\boldsymbol{x}$ of Eqs. (3.37) and (3.38), with $p = 2$, an approximation for the absolute and the relative truncation error of the original approximation is obtained.

$$\boldsymbol{\tau}^{(1)} \approx \frac{\boldsymbol{x}^{(2)} - \boldsymbol{x}^{(1)}}{1 - 2^{-s}}, \quad \tilde{\tau}_i^{(1)} = \frac{\tau_i^{(1)}}{x_i^{(1)}}, \text{ for } i = 1, \ldots, n_\mathrm{x}. \tag{3.39}$$

For the refined approximation, the estimate

$$\tilde{\tau}_i^{(p)} \approx \frac{\tau_i^{(p)}}{x_i^{(1)}} \approx \frac{1}{p^s} \cdot \frac{\tau_i^{(1)}}{x_i^{(1)}} = \frac{\tilde{\tau}_i^{(1)}}{p^s} \tag{3.40}$$

results. The magnitude of this relative error has to be smaller than the desired relative tolerance ε_rel. Solving for p yields the state-variable

[10] Another common assumption is that the solution has a "memory" and thus $c_{i,k+1}/c_{i,k} \approx c_{i,k}/c_{i,k-1}$. In step-size control for ODE solvers, the two models may be combined, see e.g. [95].

individual step refinement

$$p_i = \left\lceil k \cdot \left(\frac{|\tilde{\tau}_i^{(1)}|}{\varepsilon_{\text{rel}}} \right)^{\frac{1}{s}} \right\rceil . \tag{3.41}$$

The "safety margin" k ensures that the desired tolerance is met on the new grid. The final step refinement is chosen as the maximum among all p_i.

In the context of optimal control, a value of $k < 1$ may be convenient. Since again an OCP is solved on the new grid, the control inputs may change. Thus, the exact solution of the ODEs as well as the error of the approximation change. Due to this changing nature of the problem, it may be advantageous to refine the grid in multiple cautious iterations instead of trying to enforce the desired accuracy by a single refinement step. In the latter case, too many grid points may be added where they are not actually needed – but this fact only turns out once the refined OCP has been solved. Especially after the first iteration on the initial grid, which usually is uniformly spaced, a factor of $k \approx 0.5$ is a reasonable choice.

3.5.2 Implementation

The current approximation $\boldsymbol{x}^{(1)}$ is available from the solution of the OCP on the current grid. The approximation $\boldsymbol{x}^{(2)}$ is obtained by solving each of the two implicit steps by a Newton-type iteration. As a close initial guess, an interpolation of the current approximation is used such that in most cases, only 1 or 2 Newton steps are required. The algebraic equation solver KINSOL from the SUNDIALS suite (v. 2.5.0, released 3/2012) is used [103, 6]. It turned out to be approximately twice as fast as MATLAB's `fsolve`.

3.6 Calculating derivatives

The calculation of derivatives is a key ingredient when solving OCPs. Three main approaches are available, which are described next.

1. *Symbolic differentiation* (SD), i.e. applying the rules of derivation to the symbolic equations. This approach, which yields derivatives at machine precision, is practical only for simple and small problems. For complex functions, its application is cumbersome, even with the support of symbolic math tools. Above all, case distinctions such as if-else statements have to be handled manually. Furthermore, extremely long expressions may result from the application of the chain and the product rules. These expressions are expensive to evaluate since they require a multiple of the mathematical operations of the original function.

2. *Finite differences* (FD). A function is treated as "black box" and is evaluated at different points to identify its reaction to a change of each of its inputs. If this "perturbation" is chosen as a small, but finite value, infinitesimal derivatives can be approximated. This method is straightforward to implement, generally applicable, and fast. The accuracy obtained by first-order approximations is sufficient for most applications, and higher-order approximations may be used if a higher accuracy is required.

3. *Algorithmic differentiation* (AD) [158, 88]. This method sometimes is termed "automatic differentiation", which is misleading since automated versions of SD and FD are also available. The main idea is to apply the chain rule of derivation to the computer code of a function, i.e. to break down the function into elementary operations. Derivative information is then propagated through the code. Forward, backward and combined propagation modes

are available. The forward mode is more efficient than the backward formulation whenever the number of outputs of a function is substantially larger than the number of inputs. The backward mode excels if the effect of many inputs on few outputs has to be assessed, e.g. in shooting methods. Its main drawback is the requirement for storage, which can be relieved by "checkpointing". In this hybrid method, the code is split into segments on which shorter backward sweeps are applied.

As for SD, exact derivatives are obtained by AD (to machine precision). One problem is the fact that overhead may be introduced. Not every operation in a function propagates derivative information in every direction, which leads to unnecessary operations if no analysis of the structure of the function is performed. This problem is encountered by simple *operator overloading* (OO) approaches. These methods modify the implementation of all elementary operations such that a vector (or matrix) of derivatives is propagaged along the normal function evaluation. *Source-code transformation* (SCT) generates new code from a user-specified function that calculates the derivatives along with the function output. The implementation of such methods is more involved than OO, but the overhead can be reduced in the process of the code transformation, and other automatic code-optimisation steps may be performed.

SD is not considered here since the model functions of the system at hand are complex and lead to extremely long expressions already for the first partial derivatives. To assess the impact of the accuracy of the derivative information on the various NLP solvers, AD is used. Note that due to the custom code used to construct the Jacobian and the Hessian as described in Sec. 3.4, only the (generally dense) Jacobian and Hessian matrices of the model functions need to be obtained. If

the structure inherent in the problem is not exploited, sparse FD may be used. The idea is to find sets of mutually independent perturbation directions to reduce the number of function evaluations. The principle is described in [143, Sec. 8.1] or [37, Sec. 2.2].

3.6.1 Finite differences

The first and second partial derivatives of all model functions are required. Consider a scalar function of two variables, $f(x,y)$. The following Taylor expansions around (x,y) are used, where $f := f(x,y)$ is used to simplify the notation.

$$f(x+\delta x, y) = f + \frac{\partial f}{\partial x}\cdot\delta x + \frac{\partial^2 f}{\partial x^2}\cdot\delta x^2 + O(\|\delta x\|^3), \tag{3.42a}$$

$$f(x-\delta x, y) = f - \frac{\partial f}{\partial x}\cdot\delta x + \frac{\partial^2 f}{\partial x^2}\cdot\delta x^2 + O(\|\delta x\|^3). \tag{3.42b}$$

By solving (3.42a) for the first derivative, the common forward-difference approximation is obtained,

$$\frac{\partial f}{\partial x} = \frac{f(x+\delta x, y) - f(x,y)}{\delta x} + O(\|\delta x\|). \tag{3.43}$$

A more accurate approximation is obtained by using one more piece of information, namely by looking forward *and* backward. The formula for the central-difference (CD) approximation is obtained by subtracting (3.42b) from (3.42a), yielding

$$\frac{\partial f}{\partial x} = \frac{f(x+\delta x, y) - f(x-\delta x, y)}{2\delta x} + O(\|\delta x\|^2). \tag{3.44}$$

The error is of order $\|\delta x\|^2$ since the second-derivative term cancels in the difference of the two expansions.

An approximation for the second derivative in one variable is obtained as follows: Solve (3.42b) for $\frac{\partial f}{\partial x} \cdot \delta x$, insert this expression in (3.42a) and solve for $\frac{\partial^2 f}{\partial x^2}$. This procedure yields

$$\frac{\partial^2 f}{\partial x^2} = \frac{f(x+\delta x, y) + f(x-\delta x, y) - 2f(x,y)}{\delta x^2} + O(\|\delta x\|). \tag{3.45}$$

A more systematical approach allows the calculation of the coefficients defining an arbitrarily accurate approximation of any n-th degree derivative [75]. A MATLAB implementation is provided at [176]. The mixed partial derivatives of higher orders are approximated by applying a low-order FD scheme in direction i to a lower-order approximation in direction j. The coefficients are thus the tensor product of the two coefficient vectors. For example, the forward-difference approximation (3.43) has the coefficient vector $(0,-1,1)^T$ for the function values at $x-1$, x and $x+1$. Thus, the matrix of coefficients for a forward-difference approximation of $\frac{\partial^2 f}{\delta x \delta y}$ is

$$\begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & -1 & 1 \end{pmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}. \tag{3.46}$$

The corresponding formula is

$$\frac{\partial^2 f}{\delta x \delta y} \approx \frac{f(x+\delta x, y+\delta y) - f(x+\delta x, y) - f(x, y+\delta y) + f(x,y)}{\delta x \delta y}. \tag{3.47}$$

Note that the order of the approximation cannot be fully exploited on a physical computer, which has a finite machine precision $\varepsilon_{\text{mach}}$. The perturbation δ may not be chosen too small in order to prevent severe round-off errors. For higher-order approximations, a larger perturbation is required, which partially compromises the gain of accuracy

due to the higher order [143, Sec. 8.1]. For example, an optimal choice of δ for the forward FD scheme (3.43) is $\delta = \sqrt{\varepsilon_{\text{mach}}}$. A perfectly scaled function is assumed to derive this value. The error thus is in the order of $\sqrt{\varepsilon_{\text{mach}}}$. For the second-order approximation (3.44), the optimal perturbation is $\delta = \varepsilon_{\text{mach}}^{\frac{1}{3}}$ and the error is in the order of $\delta^2 = \varepsilon_{\text{mach}}^{\frac{2}{3}}$. Assuming a machine precision of 10^{-16}, FD achieves an accuracy of 10^{-8} in the ideal case, whereas CD generates less than three more significant digits with an accuracy of $2.15 \cdot 10^{-11}$.

Efficient approximation of first and second derivatives If first and second derivatives need to be calculated at the same time, the first derivatives can by calculated by the central-difference scheme (3.44). The forward and backward perturbations are required anyway by the second-derivative approximation (3.45). However, many NLP solvers ask for separate functions calculating the Jacobian and the Hessian matrices, such that this advantage cannot be exploited.

If central differences for the second partial derivatives are requested, the coefficient matrix

$$\begin{pmatrix} -0.5 \\ 0 \\ 0.5 \end{pmatrix} \cdot \begin{pmatrix} -0.5 & 0 & 0.5 \end{pmatrix} = \begin{bmatrix} 0.25 & 0 & -0.25 \\ 0 & 0 & 0 \\ -0.25 & 0 & 0.25 \end{bmatrix}. \tag{3.48}$$

results. However, this scheme is not desirable since it does not utilise any of the single-directed perturbations but requires four extra two-directed perturbations. More readily available information can be used by combining the forward scheme (3.46) with its backward counterpart

by taking the average of the two, yielding the coefficient matrix

$$\begin{bmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 1 & -0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix}. \tag{3.49}$$

Compared to the forward scheme (3.46), only one additional function value needs to be known, namely $f(x-\delta x, y-\delta y)$.[11] Nevertheless, this scheme represents an estimate of the second partial derivatives that is constructed from forward *and* backward information. It is thus used in the optimal-control framework if the user requests a central-difference approximation of the second derivatives.

Model sparsity The dense finite difference schemes described above are applied to calculate the Jacobian and Hessian matrices of the model function. Of course, the model function itself can have a sparse Jacobian structure. Since the model Jacobian has dimension $n_x \times (n_u + n_x)$, it usually is small and its sparsity pattern can be derived either by physical considerations or simple numerical tests. Thus, the perturbation sets for sparse FD, if applicable, can be identified by inspection.

Similarly, the sparsity structure of the model Hessian can be derived. All Lagrange multipliers are assumed to be nonzero during the analysis. Since each entry is calculated isolated from the others, all zero entries just do not need to be evaluated. This fact makes sparsity exploitation in the Hessian calculation straightforward once the sparsity pattern is known. However, since the model Hessian is a superposition of the Hessian of all state variables, the objective, the integral and the path constraints, it usually does not exhibit any sparsity. The sparsity structure of the model at hand is analysed in Sec. 4.2.

[11] The single-directed backward perturbations are required for the calculation of the diagonal elements by Eq. (3.45) anyway.

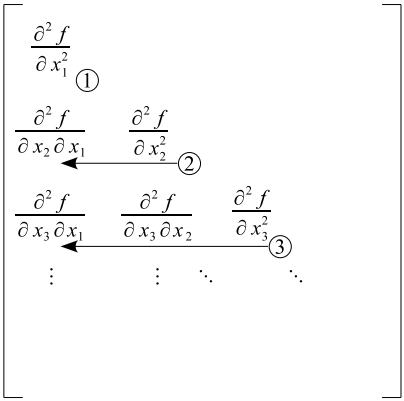

Figure 3.6: Procedure to calculate the model Hessian. Starting with the diagonal elements ensures that all data required for the off-diagonal elements is available.

Calculation procedure Since the Hessian is symmetric, only its lower triangular part has to be calculated. The calculation proceeds row by row, starting with the diagonal element, see Fig. 3.6. By this procedure, the single-directed perturbations w.r.t. the relevant variables are available for the calculation of the off-diagonal elements. Otherwise, the evaluation of the perturbed function values and the construction of the Hessian would have to be separated into two successive loops.

The first derivatives (i.e. the Jacobian) can be constructed along with the diagonal elements of the Hessian, if required.

3.6.2 Algorithmic differentiation

The most implementations of AD are available for the programming languages Fortran or C. Here, the following implementations in MATLAB are tested.

- **ADiMat** (SCT)
  http://www.sc.rwth-aachen.de/adimat/
  The source-code transformation is performed online on a prescribed server. To execute the transformed functions, an additional run-time environment is required.

- **ADMAT 2.0** (OO)
  http://www.cayugaresearch.com/admat.html

- **Automatic Differentiation for** MATLAB (OO)
  http://www.mathworks.com/matlabcentral/fileexchange/15235

- **INTLAB V6** (OO)
  http://www.ti3.tu-harburg.de/rump/intlab/

None of these tools seems to be able to successfully reduce the overhead introduced by AD. Thus, the computation times are relatively long as compared to FD. Due to its simple handling and its open-source character (BSD licence), "Automatic Differentiation for MATLAB" is used for all studies in this thesis. It also yields the fastest evaluation of first- and second-order derivatives for the problem at hand. Namely, the gradients of all model-function elements w.r.t. all state variables and control inputs is calculated in around $4.5 \cdot (n_x + n_u)$ times the time required for one function evaluation. Thus, an overhead factor of 4.5 is introduced compared to forward FD. Interestingly, the overhead is decreased to a factor of approximately 1.6 when all second partial derivatives are calculated.

The potential of achieving runtimes typical for forward FD should be analysed further using an efficient implementation of AD in Fortran or C. The considerations in this thesis focus on the influence of the accuracy of the derivatives on the behaviour of the NLP solvers

in terms of convergence and stability. Deriving or testing an efficient implementation of AD thus is beyond the scope of this work.

Details on AD can be found in dedicated books such as [158, 88]. A more compact outline focused on the application of AD in numerical optimisation is presented in [143, Sec. 8.2]. Useful introductions to AD, the latest news and progress concerning AD, and a collection of links are available at the "Community Portal for Automatic Differentiation" [1].

3.7 Dual formulation and time subdivision

OCPs of a certain structure are most efficiently solved by SQP or AS methods. For example, they are more efficient than IP solvers whenever the set of active inequality constraints can be accurately identified during the first few iterations, as it is the case for the problem at hand, see Sec. 4.4.2. On the other hand, as discussed in Sec. 3.2.1, IP solvers are capable of efficiently solving large problems. In situations where the problem is large but also features many, but "nice" inequality constraints, one faces a tradeoff between a fast convergence (SQP or AS) or a computationally efficient solution that requires many iterations to converge (IP). The problem at hand is an example for such a situation. Large time horizons have to be considered, and simultaneously, multiple inequality path constraints are present. Furthermore, the model functions are expensive to evaluate, and thus a QN approach is most promising for a fast solution.

This section presents a methodology which combines the linear increase of the runtime w.r.t. the time horizon of an IP solver with the advantages of a QN AS solver. The key idea is stated next.

Time-domain decomposition An unconstrained OCP can be divided into smaller subproblems [53]. From this subdivision of the time horizon, a two-stage problem structure results. The high-level problem solves for the optimal state at each interval boundary, whereas the individual subproblems are OCPs with a prescribed start and end state. An extension to control-constrained problems is possible [128]. Applying asynchronous iteration schemes [112] allows for an efficient parallelisation.

Here, the extension of this approach to cumulative inequality constraints is presented. Furthermore, in order to reduce the dimension of the high-level problem, a sequential solution proce-

dure is proposed. Conditions for the equivalence to the original problem are derived by means of a partially dual problem formulation, and various iterative schemes for the high-level problem are proposed and compared.

Two concepts for achieving an efficient numerical solution of the OCP that were introduced in previous sections are shown to be compatible with this procedure.

Iterative refinement, Sec. 3.5. The refinement of the approximation of the continuous-time problem can be performed simultaneously with the iterations of the high-level optimisation. Therefore, no extra NLP solutions are necessary.

Sparsity exploitation, Sec. 3.4. The subdivision of the time horizon has the goal to substantially reduce the time the NLP solver requires to solve the subproblems during each iteration. Therefore, the model evaluations become the bottleneck of the overall performance of the solution process. Exploiting the sparsity of the problem to achieve a minimum of model evaluations thus is crucial. Furthermore, the parallel calculation of the Jacobian has a more significant impact on the overall runtime for small problems. This trend is illustrated in Fig. 3.7.

The approach outlined is simple to implement and relies on any self-contained NLP solver, preferably one that implements an AS strategy. Although a specific property of the diesel-engine problem is exploited for the practical implementation developed here, the main ideas are applicable to any problem that can be cast in a similar structure. Aside from a faster solution of large-scale problems, the handling of such problems becomes more convenient. The progress and the performance of the optimisation can be assessed at intermediate stages, and

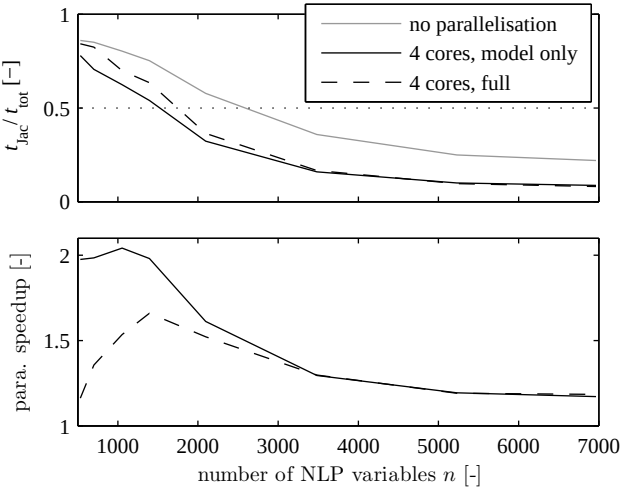

Figure 3.7: Runtime for 25 NLP iterations in SNOPT. The influence of the NLP size on the fraction of the solution time required for calculating the Jacobian (top) and the overall parallelisation speedup-factor are shown (bottom).

parameters can be adjusted according to such insights. Furthermore, if the solution of a subproblem fails, only this small problem needs to be addressed again.

3.7.1 Time-domain decomposition

In this section, the conditions are derived for the equivalence of a time-subdivided problem to the original OCP. Furthermore, two iterative schemes to achieve this equivalence are developed, as well as a way of accelerating one of them.

The primal problem defined by Eq. (3.23) is equivalent to its partially dual formulation using Lagrange multipliers for the cumulative

inequalities,

$$\min_{\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot)} \left\{ \tilde{J}\big(\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot),\boldsymbol{\pi}(\cdot)\big) = J\big(\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot),\boldsymbol{\pi}(\cdot)\big) + \boldsymbol{\lambda}^T \cdot \boldsymbol{G}\big(\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot),\boldsymbol{\pi}(\cdot)\big) \right\}, \tag{3.50}$$

where

$$J\big(\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot),\boldsymbol{\pi}(\cdot)\big) := \int_0^T L\big(\boldsymbol{u}(t),\boldsymbol{x}(t),\boldsymbol{\pi}(t)\big)\, dt, \tag{3.51}$$

$$\boldsymbol{G}\big(\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot),\boldsymbol{\pi}(\cdot)\big) := \int_0^T \boldsymbol{g}\big(\boldsymbol{u}(t),\boldsymbol{x}(t),\boldsymbol{\pi}(t)\big)\, dt. \tag{3.52}$$

The dynamic constraints and the instantaneous path constraints in Eqs. (3.23b) and (3.23d) remain unchanged, and the multipliers $\boldsymbol{\lambda} \in \mathbb{R}^{n_g}$ have to be chosen such that $\boldsymbol{G} = \hat{\boldsymbol{g}}$. Note that since any NLP solver builds the Lagrangian $\tilde{J}$ and iteratively refines the multipliers along with the primal variables, the solution of the primal problem also provides the matching values for $\boldsymbol{\lambda}$.[12]

When the full time interval $[0, T]$ is subdivided into m parts with $0 = \tau_0 < \tau_1 < \ldots < \tau_m = T$, two conditions have to be satisfied: The state trajectory $\boldsymbol{x}(t)$ has to remain continuous across the subinterval boundaries $\tau_1, \ldots, \tau_{m-1}$, and the integral inequalities of the original problem (3.23) still have to be respected. With $l = 1, \ldots, m$, $l' = 1, \ldots, m-1$ and $\tau \in [\tau_{l-1}, \tau_l)$, the subdivided OCP reads

[12] During the solution of the NLP, all constraints are adjoint to the objective by Lagrange multipliers as shown in Sec. 3.2. However, since all except the cumulative constraints are inherently time separated, they are not relevant for the methodology presented here and are thus omitted to improve readability.

$$\min_{\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot)} \sum_{l=1}^{m} J_l\big(\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot),\boldsymbol{\pi}(\cdot)\big) \tag{3.53a}$$

$$\text{s.t.} \qquad \dot{\boldsymbol{x}}(\tau) = \boldsymbol{f}\big(\boldsymbol{u}(\tau),\boldsymbol{x}(\tau),\boldsymbol{\pi}(\tau)\big), \tag{3.53b}$$

$$\boldsymbol{G}_l\big(\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot),\boldsymbol{\pi}(\cdot)\big) \leq \hat{\boldsymbol{g}}_l, \tag{3.53c}$$

$$\boldsymbol{x}(\tau)|_{\tau\nearrow\tau_{l'}} = \boldsymbol{x}(\tau)|_{\tau\searrow\tau_{l'}}, \tag{3.53e}$$

$$\sum_{l=1}^{m} \hat{\boldsymbol{g}}_l = \hat{\boldsymbol{g}}, \tag{3.53f}$$

where

$$J_l\big(\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot),\boldsymbol{\pi}(\cdot)\big) := \int_{\tau_{l-1}}^{\tau_l} L\big(\boldsymbol{u}(t),\boldsymbol{x}(t),\boldsymbol{\pi}(t)\big)\, dt, \tag{3.54}$$

$$\boldsymbol{G}_l\big(\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot),\boldsymbol{\pi}(\cdot)\big) := \int_{\tau_{l-1}}^{\tau_l} \boldsymbol{g}\big(\boldsymbol{u}(t),\boldsymbol{x}(t),\boldsymbol{\pi}(t)\big)\, dt. \tag{3.55}$$

Since the objective, the piecewise dynamic constraints in Eq. (3.53b), and $\boldsymbol{G}_l$ are separated on their respective time interval, the subproblems may be solved independently. The path constraints in Eq. (3.23d) are time-separated by nature and thus are not explicitly repeated. Achieving continuity accross the subinterval boundaries as imposed by Eq. (3.53e) and honoring the global inequality constraints of Eq. (3.53f) are within the responsibility of the high-level optimisation.

The partially dual formulation of the subdivided problem reads

$$\min_{\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot)} \left\{ \sum_{l=1}^{m} \tilde{J}_l\big(\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot),\boldsymbol{\pi}(\cdot)\big) = \underbrace{\sum_{l=1}^{m} J_l\big(\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot),\boldsymbol{\pi}(\cdot)\big)}_{=J(\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot),\boldsymbol{\pi}(\cdot))} + \sum_{l=1}^{m} \boldsymbol{\lambda}_l^T \cdot \boldsymbol{G}_l\big(\boldsymbol{u}(\cdot),\boldsymbol{x}(\cdot),\boldsymbol{\pi}(\cdot)\big) \right\}, \quad (3.56)$$

along with Eqs. (3.53b) and (3.53e). Clearly, the objective in Eq. (3.56) is equivalent to the one of the original problem stated in Eq. (3.50) if the Lagrange multipliers of all subintervals are equal, i.e. $\boldsymbol{\lambda}_l = \boldsymbol{\lambda}$ for $l = 1,\dots,m$, since in that case

$$\sum_{l=1}^{m} \tilde{J}_l = J + \boldsymbol{\lambda}^T \cdot \sum_{l=1}^{m} \boldsymbol{G}_l = J + \boldsymbol{\lambda}^T \cdot \boldsymbol{G} = \tilde{J}. \quad (3.57)$$

Thus, in order to obtain a solution set for the subproblems that is equivalent to the solution of the original problem, those values for $\hat{\boldsymbol{g}}_l$ have to be found which satisfy Eq. (3.53f) and lead to identical multipliers $\boldsymbol{\lambda}_l$.

Notion of Locality For the engine problem considered in this work, this finding can be interpreted as follows. In a driving cycle, there are segments during which it is harder to reduce the fuel consumption while maintaining a certain emission level than during others. If in spite of this fact the same stringent emission level is enforced throughout these intervals, the solver has to penalise the emissions more strongly by increasing their multiplier values. In contrast, there might be segments of the cycle during which the emissions do not even impair fuel optimality, i.e. they remain below the required level in any case. Therefore, there is no penalty for the emissions, and the multipliers are zero.

When the entire cycle is considered as one piece, these “hard” and “easy” phases cancel. In other words, the penalty for the instantaneous emission rate is the same throughout the cycle. This uniform penalisation is entirely reasonable since the time at which the emissions are produced or the fuel is saved is irrelevant.

The smaller the subintervals of the time-subdivided problem are, the more this “locality” of the problem becomes perceivable. When the intervals become too small, there might not even be a solution for some of them since the emissions cannot be reduced below the desired level (multipliers approach infinity). On the other extreme, the emissions might not impair fuel optimality at all on other intervals (multipliers are zero). The variance of the multipliers of the subproblems thus is a measure for the locality of the solution. If no locality is left, i.e. the multipliers are equal, the global solution has been found.

3.7.2 Iterative update of the multipliers

In the following, assume that the global continuity conditions (3.53e) are satisfied. Assume further that the sensitivities of the individual Lagrange multipliers $\boldsymbol{\lambda}_l$ on the corresponding inequality limits $\hat{\boldsymbol{g}}_l$, i.e. $\frac{\partial \boldsymbol{\lambda}_l}{\partial \hat{\boldsymbol{g}}_l}$, are available. These sensitivities indicate the direction for an update of the inequality limits such that the multipliers of the subintervals converge. Here, the asterisk indicates new iterates. The iteration aims to achieve

$$\boldsymbol{\lambda}_l^* = \boldsymbol{\lambda}^*, \text{ for } l = 1, \ldots, m,$$

subject to the global inequality constraint

$$\sum_{l=1}^{m} \hat{\boldsymbol{g}}_l^* = \hat{\boldsymbol{g}}. \tag{3.58}$$

The total derivatives of the multipliers are

$$d\boldsymbol{\lambda}_l = \frac{\partial \boldsymbol{\lambda}_l}{\partial \hat{\boldsymbol{g}}_l} \cdot d\hat{\boldsymbol{g}}_l + \frac{\partial \boldsymbol{\lambda}_l}{\partial \boldsymbol{x}_{l-1}} \cdot d\boldsymbol{x}_{l-1} + \frac{\partial \boldsymbol{\lambda}_l}{\partial \boldsymbol{x}_l} \cdot d\boldsymbol{x}_l, \tag{3.59}$$

where $\boldsymbol{x}_l := \boldsymbol{x}(\tau_l)$ is the state prescribed at the interval boundary l. For systems where the state trajectories are inherently time separated, the influence of the state at the boundaries on the multipliers is small, as compared to the influence of the limits $\hat{\boldsymbol{g}}_l$. In problems where the time horizon considered is several orders of magnitudes longer than the system dynamics, this assumption holds. It will be shown in Sec. 5.1.5 that the diesel-engine problem considered here exhibits this property. Furthermore, as the multipliers converge to their global values with the high-level iterations, the influence of the boundary-state terms in (3.59) vanishes anyway. The first term is thus used as a first-order approximation of the direction of the update,

$$\boldsymbol{\lambda}^* - \boldsymbol{\lambda}_l \approx \frac{\partial \boldsymbol{\lambda}_l}{\partial \hat{\boldsymbol{g}}_l} \cdot \left(\hat{\boldsymbol{g}}_l^* - \hat{\boldsymbol{g}}_l\right), \text{ for } l = 1, \ldots, m. \tag{3.60}$$

The $(m+1) \cdot n_\mathrm{g}$ unknowns, namely the new emission limits for each subinterval and the new unitary multipliers, are defined by Eqs. (3.58) and (3.60), which combine to a system of linear equations to be solved,

$$\left(\begin{array}{ccc|c} \mathbb{1}_{n_\mathrm{g}} & \cdots & \mathbb{1}_{n_\mathrm{g}} & 0 \\ \hline -\frac{\partial \boldsymbol{\lambda}_1}{\partial \hat{\boldsymbol{g}}_1} & & 0 & \mathbb{1}_{n_\mathrm{g}} \\ & \ddots & & \vdots \\ 0 & & -\frac{\partial \boldsymbol{\lambda}_m}{\partial \hat{\boldsymbol{G}}_m} & \mathbb{1}_{n_\mathrm{g}} \end{array}\right) \cdot \left(\begin{array}{c} \hat{\boldsymbol{g}}_1^* \\ \vdots \\ \hat{\boldsymbol{g}}_m^* \\ \hline \boldsymbol{\lambda}^* \end{array}\right) = \left(\begin{array}{c} \hat{\boldsymbol{g}} \\ \hline \boldsymbol{\lambda}_1 - \frac{\partial \boldsymbol{\lambda}_1}{\partial \hat{\boldsymbol{g}}_1} \cdot \hat{\boldsymbol{g}}_1 \\ \vdots \\ \boldsymbol{\lambda}_m - \frac{\partial \boldsymbol{\lambda}_m}{\partial \hat{\boldsymbol{g}}_m} \cdot \hat{\boldsymbol{g}}_m \end{array}\right). \tag{3.61}$$

Inactive inequalities On some subintervals, some of the inequality constraints may be inactive, i.e. the corresponding multipliers are zero. In this case, the limiting values for the respective inequality constraints

on each affected interval are set to the values that resulted from the preceding low-level optimisation. The sensitivites are calculated at these values, in the direction that makes the constraints active. Since the update step tries to equalise the multipliers, it will always choose more stringent limits for the respective inequality constraints for this subproblem such as to enforce nonzero multipliers.

Solving the dual form of the subproblems Instead of iterating on the inequality limits $\hat{\boldsymbol{g}}_l$, one could prescribe unitary multipliers for all subproblems in the first place. The individual subproblems would be stated in their partially dual form of Eq. (3.56), with $\boldsymbol{\lambda}_l = \boldsymbol{\lambda}^*$. This approach has the advantage that only n_g parameters – the unitary multipliers $\boldsymbol{\lambda}$ – remain in the high-level optimisation. They need to be chosen such that the global inequality constraints in Eq. (3.53f) are satisfied, i.e. the equation system

$$\hat{\boldsymbol{g}} - \sum_{l=1}^{m} \boldsymbol{G}_l = \sum_{l=1}^{m} \frac{\partial \boldsymbol{G}_l}{\partial \boldsymbol{\lambda}} \cdot (\boldsymbol{\lambda}^* - \boldsymbol{\lambda}) \tag{3.62}$$

has to be solved for the new multipliers $\boldsymbol{\lambda}^*$.

However, the sensitivity of the reduced high-level problem is higher since the full time horizon is now considered. In other words, the relation between the multipliers and the values of the integral quantities is more nonlinear, and consistent derivatives are hard to obtain. This effect is again the "tail wagging the dog" phenomenon known from single shooting methods described in Sec. 3.1.1. In that context, the problem is mitigated exactly by subdividing the time horizon into smaller intervals by applying multiple shooting. For this reason, the approach of directly solving the dual form of the subproblems is not considered further in this thesis.

Termination criteria Numerous criteria are conceivable to define convergence of the high-level iteration. The most straightforward criterion is the closeness of the multipliers $\boldsymbol{\lambda}_l$ on all subintervals l. However, due to the sensitivity of the multipliers, it might be more advisable to consider the change of the inequality limits $\hat{\boldsymbol{g}}_l$ from one iteration to the next. For the specific practical implementation presented in Sec. 3.7.5, the compliance of the global inequality limits $\hat{\boldsymbol{g}}$ has to be taken into account as well.

3.7.3 Approximate update of the multipliers

If sensitivities are not available or are expensive to calculate, an approximate iteration can be applied. It is assumed that each cumulative inequality $i \in \{1,\dots,n_\mathrm{g}\}$ may be treated independently and that its value is inversely proportional to its Lagrange multiplier to some power, ${}_i\hat{g}_l \propto {}_i\lambda_l^{-c_i}$. The left-hand index is used to denote a specific integral inequality. To simplify the tuning of this approach, a uniform power $c = c_i$, for $i = 1,\dots,n_\mathrm{g}$, is adopted here. For the update of the inequality limits,

$$ {}_i\hat{g}_l^* = {}_i\hat{g}_l \cdot \left(\frac{{}_i\lambda_l}{{}_i\lambda^*}\right)^c \tag{3.63} $$

results. Inserting this expression into Eq. (3.58) yields the new unitary multiplier

$$ {}_i\lambda^* = \left(\frac{1}{{}_i\hat{g}} \cdot \sum_{l=1}^{m} {}_i\hat{g}_l \cdot {}_i\lambda_l^c\right)^{\frac{1}{c}}, \tag{3.64} $$

which is then used to obtain the new inequality limits using Eq. (3.63).

Inactive inequalities Subintervals j for which the integral inequality constraint i is inactive do not contribute to the sum in Eq. (3.64) since

their multiplier is zero. Accordingly, the corresponding inequality value ${}_iG_j$ has to be subtracted from ${}_i\hat{g}$. The update in Eq. (3.63) is replaced by

$$ {}_i\hat{g}_j^* = x_{IR} \cdot {}_iG_j. \tag{3.65}$$

The factor x_{IR} has to be chosen such that the inequality constraint becomes active in this subinterval in the next high-level iteration. A value of 0.99 is found to be a reasonable choice for the problem at hand.

3.7.4 Acceleration of the approximate update

The information of previous iterations of the approximate update can be utilised to construct a model for the functional relation between the inequality limits and the Lagrange multipliers on each subinterval. In fact, calculating the sensitivities can be seen as deriving the first-order Taylor expansion at the current values of the inequalities. The following acceleration of the approximate update therefore is intended to reclaim the convergence rate of the original iterative scheme while avoiding the extra cost of an explicit calculation of the sensitivities during each high-level iteration. In the following, the left-hand index indicates the high-level iteration number.

A general first-order model reads

$$ \boldsymbol{G}_l \approx \boldsymbol{G}_{0,l} + \boldsymbol{A}_l \cdot \boldsymbol{\lambda}_l, \tag{3.66}$$

where $\boldsymbol{G}_{0,l} \in \mathbb{R}^{n_\mathrm{g}}$ and $\boldsymbol{A}_l \in \mathbb{R}^{n_\mathrm{g} \times n_\mathrm{g}}$. The model coefficients in $\boldsymbol{A}_l$ are identified by a linear least-squares regression (LSQ). As data for the regression, the pairs $\{\boldsymbol{G}_l, \boldsymbol{\lambda}_l\}$ of at least $1 + n_\mathrm{g}$ previous iterations are required. If more iterations are used, the data pairs are weighted by an exponential decay according to the normalised Euclidean distance of

their multipliers from the current multiplier values at iteration Q,

$$ {}_q w = \exp\left(-k_{\mathrm{dec}} \cdot \frac{\| {}_q\boldsymbol{\lambda}_l - {}_Q\boldsymbol{\lambda}_l \|_2}{\| {}_Q\boldsymbol{\lambda}_l \|_2}\right). \tag{3.67} $$

The overall procedure is provided in the following list.

1. Define how many of the most recent iterations, including the current one, are to be used (n_{acc}) and pick a value for the exponential decay, k_{dec}.

2. Calculate the weights ${}_q w$ for $q = Q - n_{\mathrm{acc}}, \ldots, Q$ according to Eq. (3.67).

3. Identify the coefficients of the linear model for each interval l and each inequality j by weighted LSQ, i.e. solve the system of linear equations

$$ (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X}) \cdot \boldsymbol{p} = \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{y} \tag{3.68} $$

for $\boldsymbol{p}$, where

$$ \begin{aligned} \boldsymbol{p} &= \left(G_{0,l,j},\ A_{l,j,1},\ \ldots,\ A_{l,j,n_{\mathrm{g}}}\right)^T, \\ \boldsymbol{W} &= \begin{bmatrix} {}_{Q-n_{\mathrm{acc}}} w & & 0 \\ & \ddots & \\ 0 & & {}_Q w \end{bmatrix}, \\ \boldsymbol{X} &= \begin{bmatrix} 1 & {}_{Q-n_{\mathrm{acc}}}\lambda_{l,1} & \cdots & {}_{Q-n_{\mathrm{acc}}}\lambda_{l,n_{\mathrm{g}}} \\ \vdots & \vdots & & \vdots \\ 1 & {}_Q\lambda_{l,1} & \cdots & {}_Q\lambda_{l,n_{\mathrm{g}}} \end{bmatrix}, \\ \boldsymbol{y} &= ({}_{Q-n_{\mathrm{acc}}} G_{l,j},\ \ldots,\ {}_Q G_{l,j})^T. \end{aligned} \tag{3.69} $$

4. To obtain the new inequality constraints and the refined estimates for the unitary multipliers, solve the following system of linear equations, which is similar to the one in Eq. (3.61).

$$\left(\begin{array}{ccc|c} \mathbb{1}_{n_g} & \dots & \mathbb{1}_{n_g} & 0 \\ \hline & & & -\boldsymbol{A}_1 \\ & \mathbb{1}_{n_g \cdot m} & & \vdots \\ & & & -\boldsymbol{A}_m \end{array}\right) \cdot \left(\begin{array}{c} \hat{\boldsymbol{g}}_1^* \\ \vdots \\ \hat{\boldsymbol{g}}_m^* \\ \hline \boldsymbol{\lambda}^* \end{array}\right) = \left(\begin{array}{c} \hat{\boldsymbol{g}} \\ \hline \boldsymbol{G}_{0,1} \\ \vdots \\ \boldsymbol{G}_{0,m} \end{array}\right). \tag{3.70}$$

3.7.5 Practical implementation

As mentioned above, the specific practical implementation presented next exploits some properties of the problem at hand and has to deal with some limitations imposed by the structure of the problem. However, the former may be featured also by similar problems, and the latter probably apply to most engineering problems.

In order to implement the two-stage optimisation, two open questions need to be answered.

1. How are the sensitivities $\frac{\partial \boldsymbol{\lambda}_l}{\partial \hat{\boldsymbol{g}}_l}$ obtained?

2. How are the global continuity conditions in Eq. (3.53e) satisfied?

These questions cannot be answered independently. The second one can be addressed only after the possible answers to the first one have been analysed.

Calculating sensitivities Under stringent conditions, the solution of an NLP can be differentiated w.r.t. arbitrary parameters of the problem by directly utilising second-order information [47]. Two main obstacles prevent the application of this elegant way of obtaining sensitivities to most practical problems such as the one considered in this work.

First, the symbolic calculation of the Hessian, i.e. the second partial derivatives, may not be possible due to the complexity of the model. Furthermore, some elements may be present that are not analytically differentiable, such as switches, absolute values or lookup maps. The presence of such elements also prevents the application of AD. Using FD to calculate the second derivatives yields an insufficient accuracy of about $\varepsilon_{\text{mach}}^{1/4} \approx 10^{-4}$. For real problems, an even lower accuracy is to be expected since a larger perturbation than the machine precision $\varepsilon_{\text{mach}}$ has to be used. A QN approximation of the Hessian, such as the popular BFGS update, is not suitable since it provides only approximate information along the direction of the last few iterations of the NLP solver.

The second problem is more severe. The conditions under which the direct sensitivity calculation is valid are almost impossible to guarantee. In particular, the active set of constraints cannot be assumed to be invariable w.r.t. perturbations of the relevant parameters of the problem. Two examples illustrate this intuition.

1. Many problems result in bang-bang type control actuation at least during certain periods of the time horizon considered. If the actuator range or some dynamics of the system are changed, the switching points are shifted and thus the active set is changed.

2. Changing the emission limits for the engine problem considered here induces a uniform tendency of some of the control inputs. For example, imposing a lower NO_x emission limit induces a generally later injection. For reasons of combustion stability or the restricted validity of the model due to such effects, the maximally allowable injection delay is limited. Thus, when the NO_x limit is lowered, this limit is reached during longer periods of time, which changes the active set of constraints.

The other way of calculating sensitivities, which is used here, is by applying FD to the NLP itself. The NLP is solved again for each parameter to be perturbed. In the worst case, the same computational cost as for the solution of the unperturbed problem is required to obtain one single sensitivity. However, only an "infinitesimally small" perturbation is necessary. Thus, the solution of the unperturbed problem can be used as a very close initial guess for the solution of the perturbed problem, which reduces the number of NLP iterations required for its solution. In addition, the active set of the perturbed solution is similar to that of the original problem, reducing the number of updates to the problem factorisation and possibly eliminating the need for complete refactorisations during the initial iterations. This "hot start" procedure is only beneficial for active-set algorithms. In contrast, interior-point methods are difficult to warm-start, as discussed in Sec. 3.2.1.

Practical tests show that solving one perturbed NLP requires around 15–40% of the time required for solving the original problem. Thus, the number of perturbations should be kept small in order not to countervail the gains of the time subdivision.

Satisfying the global continuity conditions If the evaluation of sensitivities w.r.t. problem parameters is cheap, the state variables at each subinterval boundary are introduced as additional variables for the high-level optimisation. However, the preceding section showed the sensitivity calculation to be expensive for the problem at hand. Thus, another way of handling the global continuity constraints is proposed.

A specific property of the problem considered here is exploited, namely the inherent decoupling of the state trajectories along the time axis. This decoupling mainly stems from the fact that the time horizon of the OCP is substantially larger than the characteristic time constant of the underlying dynamic system. In fact, the horizon of the OCP is in

the range of 100 s, whereas the slowest relevant system dynamics (which are imposed by the turbocharger inertia) exhibit a characteristic time of approximately 2 s. However, this property is shared by many large-scale OCPs. Of minor importance is the additional specific property that the state of the system is largely defined by the time-variable parameters $\boldsymbol{\pi}(t)$. In Sec. 5.1.5, these properties are analysed and illustrated in more detail for the problem at hand.

The inherent time separation of the state trajectories allows for a sequential solution of the subproblems along the time axis. Two neighbouring intervals have to overlap by the characteristic time of the dynamic system. As initial state for the subsequent interval, the state before the overlapping part of the preceding segment is chosen.

Extension of the iterative schemes to overlap The update of the multipliers of the integral inequality constraints has to be adapted when overlapping subintervals are considered. The following notation is used. The inequality limits for subinterval l of the subdivided problem, i.e. the original segment without the overlap phase, are represented by $\hat{\boldsymbol{g}}_l$, whereas $\underline{\hat{\boldsymbol{g}}}_l$ designates the limits for the interval including the overlap phase.

The original update introduced in Sec 3.7.2 is extended to overlapping subintervals as follows.

1. On the full interval, including the overlap phase, the sensitivities $\frac{\partial \boldsymbol{\lambda}_l}{\partial \underline{\hat{\boldsymbol{g}}}_l}$ are derived as described in Sec. 3.7.5.

2. The sensitivities $\frac{\partial \boldsymbol{\lambda}_l}{\partial \hat{\boldsymbol{g}}_l}$ for the intervals without the overlap phase are calculated from the solutions of the original and the perturbed NLPs by integrating the inequalities only up to the relevant time, i.e. to the right interval boundary. Since the multipliers $\boldsymbol{\lambda}_l$ are

constant throughout the entire subinterval, their values are also valid for the shorter interval without the overlap phase.

3. The update in Eq. (3.61) is performed using $\frac{\partial \boldsymbol{\lambda}_l}{\partial \hat{\boldsymbol{g}}_l}$, yielding the new multipliers $\boldsymbol{\lambda}^*$.

4. The difference between the new unitary multipliers and the current multipliers is used to define the new limits for the intervals with overlap. A system of linear equations of dimension n_g has to be solved for the new limits $\underline{\hat{\boldsymbol{g}}}_l^*$ for each interval l,

$$\boldsymbol{\lambda}^* - \boldsymbol{\lambda}_l = \frac{\partial \boldsymbol{\lambda}_l}{\partial \underline{\hat{\boldsymbol{g}}}_l} \cdot \left(\underline{\hat{\boldsymbol{g}}}_l^* - \underline{\hat{\boldsymbol{g}}}_l \right). \tag{3.71}$$

For the approximate update described in Sec. 3.7.3, first Eq. (3.64) is evaluated using $\hat{\boldsymbol{g}}_l$. The resulting new multipliers are used to calculate the new values of $\underline{\hat{\boldsymbol{g}}}_l$ by Eq. (3.63). For the acceleration developed in Sec. 3.7.4, the first-order approximation in Eq. (3.66) is constructed for the subintervals with and without the overlap phase. The update in Eq. (3.70) is performed for the intervals without overlap, and a subproblem-individual update similar to Eq. (3.71) yields the new limits for the intervals with overlap.

3.8 Singular arcs and regularisation

In the context of continuous-time optimal control, singular arcs are defined as time intervals on which the Hamiltonian becomes affine in at least one control input. Throughout these intervals, the control inputs are not defined by the first-order optimality conditions. From another point of view, the second-order sufficient conditions are not strictly satisfied since the second derivatives are zero. In indirect methods, a workaround is to consider time derivatives of the Hamiltonian of increasing order until the singular control input appears explicitly [79].

The presence of singular arcs introduces problems when a direct method is chosen to numerically solve the OCP. A good overview is provided in [163, Sec. 4.5]. Loosely spoken, the better the continous problem is approximated, the more exactly the singular nature of the problem is revealed. Thus, increasing the collocation order or reducing the step size can lead to unwanted effects. Namely, the Hessian of the Lagrangian, which can be seen as the discrete equivalent to the second derivative of the Hamiltonian w.r.t. the control inputs, becomes arbitrarily small. Thus, methods relying on second-order information (which virtually all efficient NLP solvers do) observe only little effect when the control trajectories are varied.

> If $\overset{*}{u}$ is a singular control, [...] the Hessian of the Lagrangian at $\overset{*}{u}$ projected onto the subspace $\{\delta u \in L^m_{\infty,2}[0,1] | g_u(\overset{*}{u})\delta u = 0\}$ is singular. Thus, from the Taylor expansion of the Lagrangian we see that, on singular arcs, small perturbations in the control have only fourth, or higher, order (very small) effects on the projected Hessian of the Lagrangian. Singularity of problem **P** will manifest itself as singularity or near-singularity in the approximating problems $\mathbf{P}_N$ for N sufficiently large. [163, p.131]

Note that in the work cited, N represents the total number of discretisation points, which is denoted by the symbol M in the thesis at hand.

The paragraph cited states that, as long as the control inputs are chosen such as to satisfy all constraints, the transcription of a singular OCP problem becomes nearly singular. The main concern is not the loss of the quadratic (or superlinear, for QN methods) convergence, but rather the presence of "spurious oscillations along singular sub-arcs".

> We believe that these spurious oscillations appear due to the accumulation of numerical errors which, on the singular sub-arcs, have very little effect on the Lagrangian because of the singularity of its Hessian. The reason this problem appears when trajectory constraints are active may have to do with the fact that the optimization algorithm chooses control iterates that cause the trajectory to follow the constraint over the active region. It may be "easier" to accomplish this using an oscillatory control. [163, p.131–132]

In [163, Sec. 4.5], a method based on the "piecewise derivative variation of the control" is proposed to regularise directly transcribed singular OCPs. Thereby, the square of the difference in the slope of each control input in two neighbouring intervals is added to the objective as a penalty term. In a second-order spline interpretation of the control inputs, this term penalises the local curvature of the control inputs.[13] The regularisation term for a scalar control input u is

$$L_{\mathrm{reg}}(u_\bullet) := c_M \cdot \dot{\mathrm{Var}}^2_{t_N}(u_\bullet) = c_M \cdot \frac{1}{2} \sum_{l=3}^{M-1} |s_{l+1} - s_l|^2, \tag{3.72}$$

where $s_l = (u_l - u_{l-1})/(t_l - t_{l-1})$ is the slope of the control-input trajectory in interval l. The summation starts at $l = 3$ since the first point is

[13] The continuous representation of the control inputs is not unique as it is the case for the state variables, which are approximated by the collocation polynomial. Often, besides quadratic splines as assumed here, cubic splines are used to construct smooth control trajectories [37, Sec. 4.7.1].

not a collocation point in Radau methods and thus, the value of the control input at this point does not have a meaning (and is usually excluded from the NLP). The factor c_M is chosen as

$$c_M := \frac{c_{\text{reg}}}{(M-3)(M-1)^2}. \tag{3.73}$$

The factor $(M-3)$ accounts for the number of summation terms, whereas $(M-1)^2$ is an approximation of the average step size applied to discretise the control inputs. This formulation scales the regularisation term according to the chosen resolution of the approximation, such that the influence of the user-specified parameter c_{reg} is invariant.

This method has the property that the solution of the regularised problem converges to the solution of the original problem for increasingly small step sizes. Since the steps between each two neighbouring collocation nodes are considered here, both the h and the p approaches for an iterative refinement of the collocation scheme are considered. It is further shown in the original literature that c_M goes to zero fast enough as $M \to \infty$ such as to ensure that the regularisation term stays bounded.

3.8.1 Implementation with exact derivatives

The terms of the regularisation (3.72) are quadratic functions of the discretised control inputs. Thus, their first and second derivatives are linear and constant, respectively. They can be calculated analytically, which is preferable to just adding the regularisation to the objective term in the model function and thus applying FD to estimate the derivatives.[14]

[14] This seemingly "easy way out" cannot be implemented in the framework developed here since all functions are assumed to occur in time-step separated form only, cf. Sec. 3.4.

One summand in (3.72) reads

$$\Delta s_l^2 := \frac{1}{2}(s_{l+1} - s_l)^2 = \frac{1}{2}\left(\frac{u_{l+1} - u_l}{h_{l+1}} - \frac{u_l - u_{l-1}}{h_l}\right)^2, \tag{3.74}$$

with $h_l := t_l - t_{l-1}$. The partial derivatives of this term, expanded in $u_\bullet$, are

$$\frac{\partial \Delta s_l^2}{\partial u_{l+1}} = u_{l+1} \cdot \frac{1}{h_{l+1}^2} + u_l \cdot \left(-\frac{h_{l+1} + h_l}{h_l h_{l+1}}\right) + u_{l-1} \cdot \frac{1}{h_l h_{l+1}}, \tag{3.75a}$$

$$\begin{aligned}\frac{\partial \Delta s_l^2}{\partial u_l} = {} & u_{l+1} \cdot \left(-\frac{h_l h_{l+1} + h_l^2}{h_l^2 h_{l+1}^2}\right) + u_l \cdot \frac{h_{l+1}^2 + 2h_l h_{l+1} + h_l^2}{h_l^2 h_{l+1}^2} \\ & + u_{l-1} \cdot \left(-\frac{h_{l+1}^2 + h_l h_{l+1}}{h_l^2 h_{l+1}^2}\right),\end{aligned} \tag{3.75b}$$

$$\frac{\partial \Delta s_l^2}{\partial u_{l-1}} = u_{l+1} \cdot \frac{1}{h_l h_{l+1}} + u_l \cdot \left(-\frac{h_{l+1} + h_l}{h_l^2 h_{l+1}}\right) + u_{l-1} \cdot \frac{1}{h_l^2}. \tag{3.75c}$$

Each discretisation point takes the role of $l+1$, l, and $l-1$ once (the special cases for $l = 1, 2, 3, M-1, M$ are not shown here). Thus, the derivative of (3.72) w.r.t. the control input at any point l is a sum of the three partial derivatives in (3.75) with its indices shifted accordingly.

The result is

$$\begin{aligned}\frac{\partial \dot{\mathrm{Var}}^2_{t_N}(u_\bullet)}{\partial u_l} =& u_{l-2}\cdot\frac{1}{h_{l-1}h_l} + u_{l-1}\cdot\left(-\frac{h_l+h_{l-1}}{h_{l-1}h_l^2} - \frac{h_{l+1}^2+h_lh_{l+1}}{h_l^2h_{l+1}^2}\right)\\ &+ u_l\cdot\left(\frac{1}{h_l^2} + \frac{h_{l+1}^2+2h_lh_{l+1}+h_l^2}{h_l^2h_{l+1}^2} + \frac{1}{h_{l+1}^2}\right)\\ &+ u_{l+1}\cdot\left(-\frac{h_lh_{l+1}+h_l^2}{h_l^2h_{l+1}^2} - \frac{h_{l+2}+h_{l+1}}{h_{l+1}^2h_{l+2}}\right)\\ &+ u_{l+2}\cdot\frac{1}{h_{l+1}h_{l+2}}.\end{aligned} \tag{3.76}$$

This term has to be added to the gradient of the objective.

The Hessian of Δs_l^2 is

$$\frac{\partial^2\Delta s_l^2}{\partial^2(u_{l-1},u_l,u_{l+1})} = \begin{bmatrix} \frac{1}{h_l^2} & -\frac{h_{l+1}+h_l}{h_l^2h_{l+1}} & \frac{1}{h_lh_{l+1}} \\ -\frac{h_{l+1}+h_l}{h_l^2h_{l+1}} & \frac{h_{l+1}^2+2h_lh_{l+1}h_l^2}{h_l^2h_{l+1}^2} & -\frac{h_{l+1}+h_l}{h_lh_{l+1}^2} \\ \frac{1}{h_lh_{l+1}} & -\frac{h_{l+1}+h_l}{h_lh_{l+1}^2} & \frac{1}{h_{l+1}^2} \end{bmatrix}. \tag{3.77}$$

Diagonally shifting this matrix by one and two increments in both directions and collecting terms yields the entries of the Hessian of the full regularisation term, centered at u_l.

$$\frac{\partial^2\dot{\mathrm{Var}}^2_{t_N}(u_\bullet)}{\partial^2(u_{l-1},u_l,u_{l+1})} = \begin{bmatrix} * & * & \frac{1}{h_lh_{l+1}} \\ * & \frac{1}{h_l^2}+\frac{h_{l+1}^2+2h_lh_{l+1}+h_l^2}{h_l^2h_{l+1}^2}+\frac{1}{h_{l+1}^2} & -\frac{h_{l+1}+h_l}{h_lh_{l+1}^2}-\frac{h_{l+2}+h_{l+1}}{h_{l+1}^2h_{l+2}} \\ \frac{1}{h_lh_{l+1}} & -\frac{h_{l+1}+h_l}{h_lh_{l+1}^2}-\frac{h_{l+2}+h_{l+1}}{h_{l+1}^2h_{l+2}} & \circ \end{bmatrix}. \tag{3.78}$$

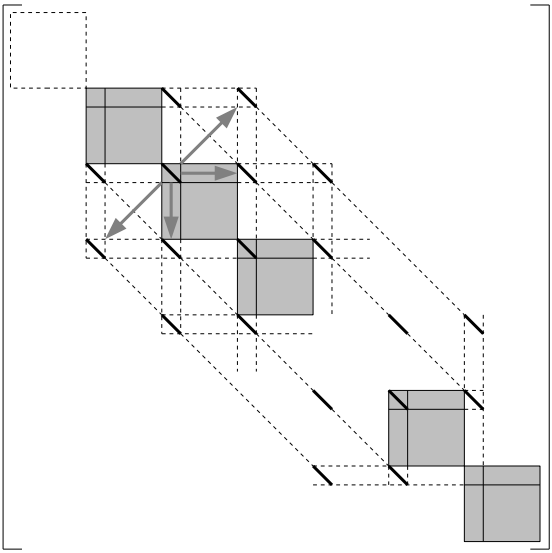

Figure 3.8: Structure of the Hessian of the Lagrangian, including the regularisation terms (bold diagonal entries). The arrows indicate which off-diagonal elements are introduced along with the diagonal block associated with each discretisation point.

The entries denoted by $*$ are calculated already by the previous step, while the entry replaced by $\circ$ is calculated by the successive step. This procedure is illustrated in Fig. 3.8. The grey blocks represent the perfectly block-diagonal Hessian of the Lagrangian of the non-regularised NLP. The contributions of the regularisation are just superposed on this original Hessian.

Three notes concerning the change of the structure of the Hessian introduced by the regularisation follow.

- The regularisation breaks the separability of the Lagrangian, visually indicated by the elements outside the diagonal blocks. The effects on the performance of the direct linear solver MUMPS

used in IPOPT is found to be negligible since the Hessian retains a banded structure.

- Due to the explicit calculation of the contribution of the regularisation to the Hessian, the partitioned QN update proposed in Sec. 3.4.3 can still be applied. The update is performed using the formulation for the original problem, and the entries stemming from the regularisation are superposed afterwards.

- As mentioned above, MATLAB stores sparse matrices in the compressed column format. In this format, all three vectors (start of column, row indices, and data) have to be modified when a nonzero element is added to a column which is not the last nonzero column. The construction scheme illustrated in Fig. 3.8 changes n_u columns that are separated from the current highest-index nonzero column by $n_\mathrm{x} + n_\mathrm{u}$ columns, as well as the n_u columns added most recently. Since only "nearby" columns are changed and since the model evaluations by far outweigh the computational expense to construct the Hessian matrix, the resulting overhead is not perceivable for the problem at hand. However, for maximum computational efficiency, the sparse Hessian should be constructed in a column-stepping order instead of a discretisation-point loop.

3.9 Multiple-shooting imitation mode

Multiple shooting can be considered a specific form of regularisation. In fact, the control inputs are usually assumed to be piecewise constant, but the ODE is solved to a prescribed accuracy and thus on an arbitrarily fine grid. In contrast to the regularisation approach described in the previous section, fast changes of the control inputs are not penalised, but they are allowed only at a specific and relatively low frequency.

Direct collocation can be easily adapted such as to imitate a shooting method. Namely, linear constraints can be appended to the problem formulation which impose a piecewise constant or a piecewise linear control. This "multiple-shooting imitation mode" (MSIM) does not bring with it the benefit of multiple shooting that the ODE is always integrated with a prescribed accuracy, which eliminates the need for an iterative mesh refinement. However, the collocation order on the control-discretisation intervals can be chosen as high as to always ensure a certain minimum accuracy of the ODE solution. Furthermore, the simultaneous nature of the solution process with all its advantages is preserved.[15]

Note that it is possible to construct infeasible sets of constraints when path constraints and MSIM are combined. For example, if a prescribed smooth state trajectory has to be followed, this time-variable constraint possibly cannot be satisfied by a piecewise constant control. In such cases, certain control inputs may have to be excluded from the MSIM constraints and thus are allowed to still be chosen individually at each collocation point. For the engine problem considered here, the

[15] A variant of MSIM is presented in [107]. A forward evaluation of the ODEs on the prescribed collocation grid is performed during each iteration to always enforce strict feasibility w.r.t. the dynamic constraints. The sensitivities are propagated along the solution. For problems with few active constraints besides the eliminated dynamic constraints, it is found that substantially larger steps are taken, resulting in fewer NLP iterations being required for convergence to the solution.

fuel injection cannot be included in MSIM due to the requirement of following the desired load torque over time, see Sec. 5.1.1.

The piecewise-constant as well as the piecewise-linear formulation of MSIM can be combined with the regularisation described in Sec. 3.8. In the case of piecewise-constant control, steps are the only way the control inputs may change. This characteristic has the advantage that large steps of the control inputs can be realised if they actually are optimal. The application of a regularising penalty to these steps, at which the curvature of the control-input trajectories strongly changes twice, reveal to the optimisation which changes are worthwhile and which are not. Therefore, an effective regularisation can be expected.

Piecewise constant control inputs The implementation of the constraints enforcing a piecewise constant control is presented for Radau collocation only. On any collocation interval k with order s, the constraints

$$\boldsymbol{g}_{\mathrm{MSIM},lk} = \boldsymbol{u}_l - \boldsymbol{u}_{k+1} = 0, \quad l = l(k,i), \quad i = 1,\dots,s-1 \tag{3.79}$$

are imposed. These constraints enforce that the control inputs at all collocation points inside the interval are the same as those at the right boundary of the interval. This choice is in accordance with the Radau collocation scheme, where the right but not the left boundary is a collocated point.

The first partial derivatives of the additional constraints read

$$\frac{\partial \boldsymbol{g}_{\mathrm{MSIM},lk}}{\partial \boldsymbol{u}_{k+1}} = -\mathbb{1}, \quad \frac{\partial \boldsymbol{g}_{\mathrm{MSIM},lk}}{\partial \boldsymbol{u}_l} = \mathbb{1}. \tag{3.80}$$

The second derivatives vanish, and thus these constraints do not contribute to the Hessian of the Lagrangian.

Piecewise linear control inputs For a piecewise linear representation of the control inputs, the values at the collocation points inside interval k are

$$\begin{aligned}\boldsymbol{u}_l &= \boldsymbol{u}_k + \frac{t_l - t_{k-1}}{t_{k+1} - t_{k-1}} \cdot (\boldsymbol{u}_{k+1} - \boldsymbol{u}_k) \\ &= \boldsymbol{u}_k \cdot \left(1 - \frac{t_l - t_{k-1}}{h_{k-1}}\right) + \boldsymbol{u}_{k+1} \cdot \frac{t_l - t_{k-1}}{h_{k-1}} \qquad (3.81) \\ &= \boldsymbol{u}_k \cdot (1 - c_l) + \boldsymbol{u}_{k+1} \cdot c_l. \qquad (3.82)\end{aligned}$$

Here, c_l denotes the normalised position of collocation node l w.r.t. the collocation interval to which it belongs. These normalised collocation points are readily available since the calculation of the nodes is performed for a normalised interval anyway, see Sec. 3.3.1.

The constraints enforcing piecewise linear control inputs thus read

$$\boldsymbol{g}_{\mathrm{MSIM},lk} = \boldsymbol{u}_l - (1-c_l)\cdot\boldsymbol{u}_k - c_l \cdot \boldsymbol{u}_{k+1} = 0, \quad l = l(k,i), \quad i = 1,\ldots,s-1. \tag{3.83}$$

The first partial derivatives are

$$\frac{\partial \boldsymbol{g}_{\mathrm{MSIM},lk}}{\partial \boldsymbol{u}_{k+1}} = -c_l \cdot \mathbb{1}, \quad \frac{\partial \boldsymbol{g}_{\mathrm{MSIM},lk}}{\partial \boldsymbol{u}_k} = (1-c_l)\cdot\mathbb{1}, \quad \frac{\partial \boldsymbol{g}_{\mathrm{MSIM},lk}}{\partial \boldsymbol{u}_l} = \mathbb{1}, \tag{3.84}$$

while, as for the piecewise-constant case, the second derivatives vanish.

4 Optimal control of diesel engines

This chapter describes the optimal control problem (OCP) for diesel engines. The problem formulation presented here is the final result of many trials and ensures a reliable and numerically stable solution. Where appropriate, alternative formulations are indicated in the text. After the formulation of the problem, some specific aspects are analysed in detail, such as the sparsity structure of the engine models, the transient driving cycle used to test the various algorithms described in Chapter 3, and the handling of drag phases in the dynamic optimisation. Finally, an alternative for the "one-step" approach is proposed. Instead of deriving an engine model that is valid over the full actuator ranges and that can predict the engine behaviour for all possible transient events, an iterative dynamic model refinement for a prescribed driving profile is developed. Not just a single OCP is solved, but a sequence of band-constrained OCPs is constructed. Between the solution of two consecutive OCPs, the model is fine-tuned around the current state trajectories.

4.1 Problem formulation

The OCP of diesel engines can be cast in the form of the general OCP in Eq. (3.2). The objective is to minimise the cumulative fuel consumption, i.e. $L = \overset{*}{m}_{\text{fuel}}$ in (3.2a). The state variables are defined by the air-path model structure chosen, and the control inputs comprise the air-path actuators, the fuel injected per cylinder and combustion cycle, and the signals that control the combustion. The engine speed, its time derivative, and the desired load torque are introduced as time-variable parameters. Therefore, in the most general case, the vectors of the state variables, the control inputs and the time-variable parameters read

$$\boldsymbol{x} = \begin{pmatrix} p_{\text{IM}} \\ p_{\text{EM}} \\ \omega_{\text{TC}} \\ p_1 \\ p_4 \\ \vartheta_{\text{IM}} \\ x_{\text{BG}} \\ \hat{\vartheta}_{\text{c}} \\ \vartheta_{\text{IM,in}} \end{pmatrix}, \quad \boldsymbol{u} = \begin{pmatrix} u_{\text{VGT}} \\ u_{\text{EGR}} \\ u_{\text{EF}} \\ m_{\text{fcc}} \\ \varphi_{\text{SOI}} \\ p_{\text{rail}} \end{pmatrix}, \quad \boldsymbol{\pi} = \begin{pmatrix} N_{\text{eng}} \\ \dot{N}_{\text{eng}} \\ \hat{T}_{\text{load}} \end{pmatrix}. \tag{4.1}$$

The fuel mass-flow is simply $\overset{*}{m}_{\text{fuel}} = m_{\text{fcc}} \cdot N_{\text{eng}}/120 \cdot n_{\text{cyl}}$.

The dynamic constraints (3.2b) represent the air-path model. The initial state is fixed by (3.2c) to a prescribed value that is obtained by simulation. The prescription of an initial state is especially important for the thermal dynamics. If no initial values are prescribed for these two state variables, the optimisation chooses them as the maximally allowable casing temperature and the lowest allowable dynamic intake temperature. This choice enables a low fuel consumption for two reasons. On the one hand, due to the higher enthalpy available to the

turbine, the VGT can be operated at a generally more open position, which reduces the gas-exchange losses. On the other hand, the ignition can be advanced to further improve the fuel economy because the NO_x emissions are low due to the low temperature of the cylinder charge.

Another option is to precalculate the trajectories of the two thermal state variables by a forward simulation and to represent them as additional time-variable parameters in the OCP. This procedure is plausible since the temperature of the exhaust gas is only marginally changed by the optimisation of the control-input trajectories. Moreover, the state vector is reduced by two entries, yielding a smaller NLP after the transcription. However, preliminary tests show that the NLP solvers require the same number of iterations to solve the slightly smaller problem. In fact, the number of the degrees of freedom is not affected. Furthermore, the time for the solution is only reduced in the order of 10%. For these reasons, the consistent formulation, which includes the thermal dynamics as state variables in the OCP, is preferred and is exclusively used here.[1]

The cumulative pollutant emissions are limited by the integral inequality constraints (3.2d), i.e. $\boldsymbol{g} = (\overset{*}{m}_{\mathrm{NOx}}, \overset{*}{m}_{\mathrm{soot}})^T$. The absolute limits $\hat{\boldsymbol{g}}$ are calculated from the desired brake-specific values by multiplication with the integral of the nonnegative segments of the engine power $P_{\mathrm{eng}} = N_{\mathrm{eng}} \cdot \pi/30 \cdot \hat{T}_{\mathrm{load}}$.

The desired load torque is imposed as a time-variable lower bound by a path constraint. The rationale for this formulation and a detailed description are provided in Sec. 4.4 below. One additional path constraint may be added to the problem formulation in some cases. It limits the AFR by imposing a lower boundary. The most general vector of path constraints in (3.2e) therefore reads $\boldsymbol{c} = (\hat{T}_{\mathrm{load}} - T_{\mathrm{load}}, \lambda_{\mathrm{AFR,min}} - \lambda_{\mathrm{AFR}})^T$.

[1] For receding-horizon or model-predictive control, where only a small time horizon is considered, the thermal dynamics can possibly be neglected entirely.

The simple bounds on the state variables (3.2f) and on the control inputs (3.2g) represent various constraints. First, the physical actuator ranges need to be respected. The fuel mass is limited from below by zero, and from above by the maximum injection quantity allowed for the engine at hand. Second, mechanical limits are imposed on certain state variables such as the maximum turbocharger speed and maximum pressures in the intake and exhaust manifolds. For the general lower limits, as well as the upper limits for the state variables which need not be limited for mechanical reasons, physically plausible values are used. These "global" limits are also used to scale the problem in the optimal control framework, see Appendix D. Finally, model-validity bounds are imposed on certain variables. For all control inputs except for the VGT position and the fuel mass, the maximal variations used during the model identification are used as limits. However, these limits are found to be hardly ever active at the optimal solution. The VGT, which largely defines the slow turbocharger dynamics, cannot be restricted by limits obtained during stationary operation of the engine.

Basic problem formulation for engine A This engine does not have an EGR system, and for the purpose of optimal control, the air-path model structure without the AFM and ATS restrictions is used. Thus, the four state variables from p_1 to x_{BG} are not present in the model, and $\vartheta_{\mathrm{IM,in}}$ is replaced by ϑ_{IM}, which is defined by Eq. (2.51) in this case. Furthermore, the two control inputs u_{EGR} and u_{EF} are not present.

Since a model for the soot emissions is available for these engines, this emission species is limited by an integral constraint. Conversely, no limit on the AFR is required, and thus the second path constraint does not apply.

Basic problem formulation for engine B with EGR The full state vector presented in Eq. (4.1) is required to represent the air path of this engine. Similarly, all control inputs apply. However, since no satisfactory soot model could be derived for this engine, the integral constraint for this emission species is not included in the problem formulation. To account for the most prominent influence on the soot emissions, the AFR is limited by the corresponding path constraint described above.

The rail pressure defines a tradeoff between the soot emissions and the NO_x emissions. If the rail pressure is too low, the combustion is slow and incomplete, which causes high soot emissions. Conversely, this slow and cool combustion leads to less thermal formation of NO_x. The negative effect of a low rail pressure on the combustion efficiency is almost outweighed by the reduced power consumption of the high-pressure pump of the common-rail system. Thus, if the soot emissions are not modelled and limited, there is no tradeoff for the rail pressure, and the optimisation would always set it to its lower bound. For this reason, the rail pressure is excluded as a control input, and the values from the current engine calibration are used instead.

Dynamic loops For the engine with EGR, the optimisation terminates prematurely for all problem instances tested. Either a QP subproblem of some outer iteration is infeasible, or the algorithm just diverges. The reason is the additional dynamic loop introduced by the EGR system. It interacts with the dynamic loop of the turbocharger and thus introduces a high degree of nonlinearity to the dynamic system. A similar observation is reported in [89]. To resolve the problem, the authors propose to break the loops and to apply a homotopy to close them again.

For the problem at hand, a more straightforward solution is implemented. The control inputs and the state variables are restricted to

within a small region around the initial trajectories. After solving this problem, the resulting trajectories are used as a new initial guess. This procedure is repeated until the solution lies within the restricted region. An SQP method is able to efficiently solve this sequence of related problems, see Sec. 5.1.2.

4.1.1 Initialisation

A good initial guess is crucial for the efficient and reliable solution of the OCP. Most NLP solvers provide a recovery mode if one of the QP subproblems is infeasible. However, for the problem at hand, none of these strategies is able to successfully recover whenever a single QP is infeasible. Even if no EGR system is present, the dynamic couplings in the system, which introduce additional nonlinearity, inhibit the solver from rediscovering the "feasible path". Obviously, also the first QP has to be feasible, which is ensured by a reasonable initialisation of the control inputs and a respective consistent initialisiation of the state trajectories and the integral constraints.

The profiles of the engine speed and the desired load torque $\hat{T}_{\mathrm{load}}$ are derived from a transient driving cycle as explained in Sec. 4.3 below. The control inputs are initialised using the signals generated by the engine control unit (ECU) when this cycle is driven. The initialisation thus represents the actual control structure implemented on the ECU. Basically, a forward simulation of the model yields the initial guess for the state trajectories and the traces of the cumulative fuel consumption and pollutant emissions.

However, two problems arise. First, usually only the causal model formulation is available, i.e. the model requires the fuel injection as a control input and provides the net torque generated by the engine as an output. Second, the transient cycles are as aggressive such as the engine reaches very low AFRs. During the homologation procedure the

engine is allowed to not always exactly follow the desired load-torque profile, such as when a lower limit for the AFR becomes active, which is commonly termed the "smoke limiter". In order to enable a comparison of the optimal solution to the initial guess, this AFR limit has to be considered during the initialisation. The procedure outlined next solves both problems.

1. Initialise $m_{\mathrm{fcc}}(t)$ by the ECU value measured when driving the cycle. Set a lower limit $\lambda_{\mathrm{AFR,min}}$ for the AFR, which usually is a value around 1.3.

2. Perform a forward simulation of the model including the AFR limit. During the simulation, apply the trajectories recorded from the ECU for all other control inputs except for the fuel injection. The output of the simulation comprises the trajectories of the actually injected fuel mass $\hat{m}_{\mathrm{fcc}}$, which is possibly limited by the AFR limitation and the actually achieved load torque T_{load}.

3. Perform a second simulation. This time, disable the AFR limiter (e.g. by just setting $\lambda_{\mathrm{AFR,min}} = 1$) and perturb the fuel injection by a constant offset, $\hat{m}^{+}_{\mathrm{fcc}}(t) = \hat{m}_{\mathrm{fcc}}(t) + \delta m_{\mathrm{fcc}}$. The offset can be chosen as a small fraction of the maximal value over the cycle under consideration, i.e. $\delta m_{\mathrm{fcc}} = 0.01 \cdot \max_t \hat{m}_{\mathrm{fcc}}(t)$. The output of this simulation provides the load torque T^{+}_{load} that corresponds to this increased fuel injection.

4. Update the fuel injection by the time-resolved first-order sensitivity of the torque w.r.t. the fuel injection,

$$m_{\mathrm{fcc}}(t) = \hat{m}_{\mathrm{fcc}}(t) + \delta \hat{m}_{\mathrm{fcc}} \cdot \frac{\hat{T}_{\mathrm{load}} - T_{\mathrm{load}}}{T^{+}_{\mathrm{load}} - T_{\mathrm{load}}}. \tag{4.2}$$

5. Repeat steps 2 to 4 until the change of the fuel injection during the update step vanishes. Even if the initialisation is poor, only 2-3 iterations are sufficient. This fast convergence stems from the fact that the torque generated by the engine is an almost perfect affine function of the fuel injection.

The load-torque trajectory derived by this procedure is used as the desired load torque for the OCP. The state variables as well as the cumulative fuel consumption and the pollutant emissions are initialised by the trajectories resulting from a last simulation of the model according to step 2.

If an inverted model formulation according to Sec. 2.3 is available as well, this model can be used to derive a close guess to start the iterative procedure just described. However, drag phases require a special treatment, since no value for the load torque is prescribed but just the fuel injection is cut off. Since the inverted and the causal formulations never yield exactly the same results, the final initialisation should always be derived by the causal model, which is used in the OCP. Only by this procedure, a feasible initialisation can be ensured.

4.1.2 Possible extensions of the problem formulation

Various extensions of the problem formulation are possible. For example, the maximum in-cylinder pressure could be limited. However, a model for this quantity had to be derived first. Preliminary tests indicate that a simple regression using the start of combustion, the engine speed, the rail pressure and the ignition delay is able to predict the peak pressure in the cylinder with a relative error magnitude of below 5% for all variations performed. Especially the critically high peak pressures are accurately reproduced.

Another important step is to include the aftertreatment system (ATS) in the model and to also optimise the corresponding control in-

puts. By this extension, the operation of the engine and the ATS could be perfectly matched to fully exploit the potential of these interacting systems. A first step towards this goal was undertaken in a master thesis supervised by the author [S.11]. Several critical points were identified during that work. Above all, the identification of the model parameters of the ATS models require a sophisticated and highly customised procedure. Due to time limitations and upon request of the industrial partner, the emphasis was placed on extending the engine system to include EGR rather than on developing optimisation-oriented models for the ATS.

As an intermediate step towards an inclusion of the ATS in the optimisation, the instantaneous emissions could be limited. This limitation can be achieved by simply introducing additional nonlinear path constraints. The limits could be chosen as a function of the engine operating point or even depending on its history. Therefore, the main characteristics and dynamic properties of the ATS could be accounted for in the optimisation without actually requiring a model.

4.2 Model sparsity

During the numerical solution of the OCP by direct transcription, the model is linearised along its current trajectory. This principle is referred to as "quasi-linearisation" [26, 111]. The matrices resulting from this linearisation are used to construct the Jacobian of the NLP. The model is characterised by the state update, i.e. the ODEs, and the output function

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{x}), \tag{4.3a}$$

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{u}, \boldsymbol{x}). \tag{4.3b}$$

Its linearisation around the point $\{\boldsymbol{u}_0, \boldsymbol{x}_0\}$ is defined by four Jacobian matrices,

$$\dot{\boldsymbol{x}} \approx \boldsymbol{f}(\boldsymbol{u}_0, \boldsymbol{x}_0) + \left.\frac{\partial \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{x})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}_0, \boldsymbol{x}_0} \cdot (\boldsymbol{u} - \boldsymbol{u}_0) + \left.\frac{\partial \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{u}_0, \boldsymbol{x}_0} \cdot (\boldsymbol{x} - \boldsymbol{x}_0), \tag{4.4a}$$

$$\boldsymbol{y} \approx \boldsymbol{g}(\boldsymbol{u}_0, \boldsymbol{x}_0) + \left.\frac{\partial \boldsymbol{g}(\boldsymbol{u}, \boldsymbol{x})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}_0, \boldsymbol{x}_0} \cdot (\boldsymbol{u} - \boldsymbol{u}_0) + \left.\frac{\partial \boldsymbol{g}(\boldsymbol{u}, \boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{u}_0, \boldsymbol{x}_0} \cdot (\boldsymbol{x} - \boldsymbol{x}_0). \tag{4.4b}$$

In the context of linear system theory and control systems, this system is commonly rearranged and the matrices named such as to yield

$$\delta \dot{\boldsymbol{x}} = \boldsymbol{B} \cdot \delta \boldsymbol{u} + \boldsymbol{A} \cdot \delta \boldsymbol{x}, \tag{4.5a}$$

$$\delta \boldsymbol{y} = \boldsymbol{D} \cdot \delta \boldsymbol{u} + \boldsymbol{C} \cdot \delta \boldsymbol{x}. \tag{4.5b}$$

By applying direct transcription to the system, the right-hand function $\boldsymbol{f}(\boldsymbol{u}, \boldsymbol{x})$ defining the ODEs and the output function $\boldsymbol{g}(\boldsymbol{u}, \boldsymbol{x})$ occur in a time-separated way only, cf. Sec. 3.4.1. Thus, the model Jacobians $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ need to be evaluated at all grid points of the discretisation

scheme applied. The engine model without EGR and without the additional restrictions exhibits the following structure of these matrices.

		control inputs				state variables				
		u_{VGT}	φ_{SOI}	p_{rail}	m_{fcc}	p_{IM}	p_{EM}	ω_{TC}	ϑ_{IM}	$\hat{\vartheta}_{\mathrm{c}}$
state variables	p_{IM}		*	*	⊛	⊛	⊛	⊛	⊛	
	p_{EM}	⊛	*	*	⊛	⊛	⊛	⊛	⊛	⊛
	ω_{TC}	⊛	*	*	⊛	⊛	⊛	⊛	⊛	⊛
	$\hat{\vartheta}_{\mathrm{c}}$					⊛		⊛	⊛	
	ϑ_{IM}		*	*	⊛	⊛		⊛	⊛	⊛
model outputs	$\overset{*}{m}_{\mathrm{fuel}}$				⊛					
	$\overset{*}{m}_{\mathrm{NO_x}}$		*	*	⊛	*	*	*	*	*
	$\overset{*}{m}_{\mathrm{soot}}$		*	*	⊛	*	*	*	*	
	T_{load}		*	*	⊛	⊛	⊛	*	*	

(4.6)

The entries marked by an asterisk are non-zero for normal operation. The additionally encircled entries remain non-zero even during drag phases. If all rows would not be full, the number of model evaluations to calculate all partial derivatives could be reduced. The sparse non-zero entries could be moved to the zero locations to “squeeze” the full model Jacobian in Eq. (4.6) into fewer columns. For grid points in drag phases, the partial derivatives w.r.t. the SOI and the rail pressure can be omitted.

The sparsity pattern of the model can be verified by physical considerations. The VGT position only has a direct influence on the exhaust-manifold pressure by changing the mass flow through the turbine, and on the turbocharger speed by changing the turbine efficiency. The fuel mass injected and the combustion controls SOI and rail pressure affect

the emissions and the combustion efficiency, which in turn changes the load torque produced as well as the temperature of the exhaust gas. Since the turbocharger speed influences the mass flows and the efficiencies of the compressor and the turbine, it affects all variables.

If a less "dense" model formulation was used, the number of non-zero entries would remain about the same, but the number of NLP variables in the transcribed problem would increase. For example, if a receiver is introduced between the compressor and the intercooler, the turbocharger speed does not influence p_{IM} directly, but indirectly by the pressure in the newly introduced volume. Similarly, if the temperature in the exhaust manifold is introduced as a state variable, several relations in the model are decoupled and the model Jacobian becomes larger, but also sparser.

Figure 4.1 shows the sparsity pattern of the Jacobian of the three different model structures used for engine B. The sparsity pattern of the version without EGR and without the additional restrictions for the AFM and the ATS is similar to the one presented in Eq. (4.6). Note, however, that the rail pressure is excluded as control input and that the soot mass-flow is replaced by the AFR. When the restrictions are included, there still is a full row in the Jacobian, namely the one of the turbocharger speed. Since the pressure ratios over the compressor and the turbine influence the respective flows and enthalpies, the pressure before the compressor and the one after the turbine directly affect the turbocharger speed as well.

In contrast, the model with EGR has a sparsity pattern that allows a reduction of the number of function evaluations required to construct the Jacobian. The exhaust flap only affects the pressure after the turbine, and the introduction of the temperature in the intake manifold decouples the direct influence of the thermal state of the intake manifold on several dynamics and outputs. The columns of the VGT and of the

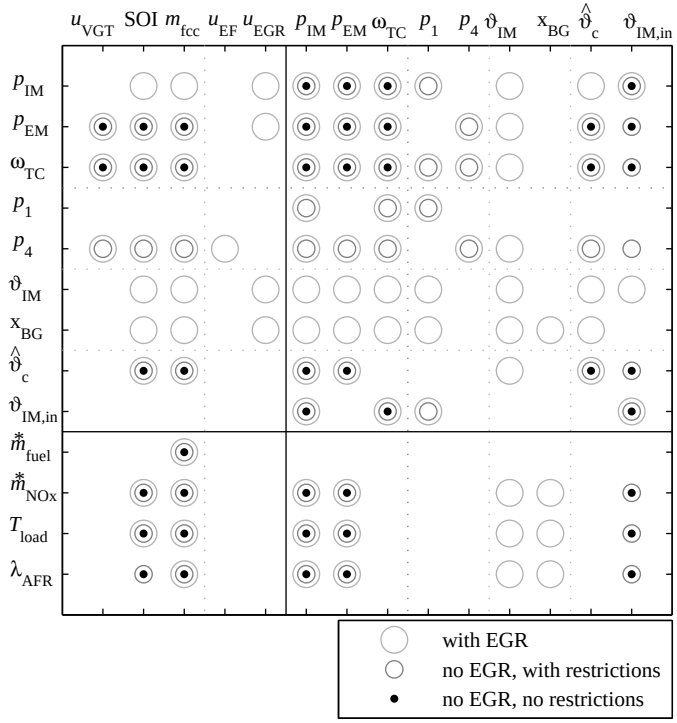

Figure 4.1: Sparsity structure of the Jacobian matrices for all three model versions, i.e. the version without and with the restrictions representing the AFM and ATS for an engine without EGR, as well as the model for an engine with EGR.

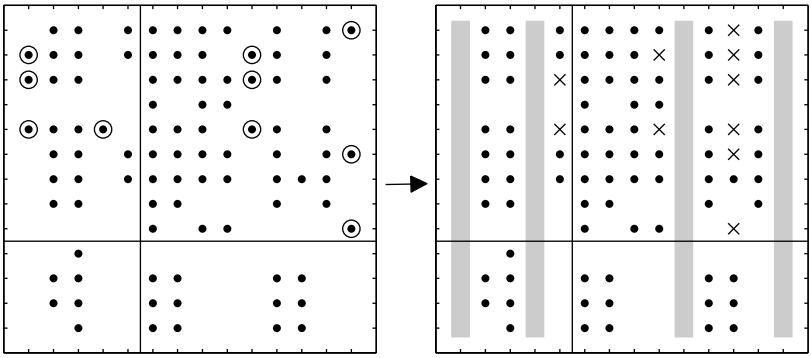

Figure 4.2: Sparsity structure of the Jacobian matrix for the model with EGR (left). The encircled entries can be moved horizontally to the locations marked by the crosses in the right-hand plot. In total, four model evaluations can be saved.

pressure after the turbine are sparsely populated anyway. The non-zero entries of the four said columns can thus be moved to free spaces within the other columns as illustrated in Fig. 4.2. Four model evaluations can be saved for the construction of the model Jacobian, which mitigates the extra computational time required for the model evaluations during the solution of an OCP for an engine with EGR. However, this fact only holds for a quasi-Newton solver. Since the model Hessians remain dense, see below, the number of model evaluations cannot be reduced for exact Newton methods. Furthermore, the number of NLP variables remains substantially larger.

Hessian structure Figure 4.3 shows the sparsity pattern of the Hessian matrices of all state variables and outputs. Since the Hessian of the turbocharger speed is completely full, the time-separable terms of the NLP Hessian (3.33) are full. No sparsity can thus be exploited when the Hessian of the Lagrangian is calculated. Note that efficient AD codes

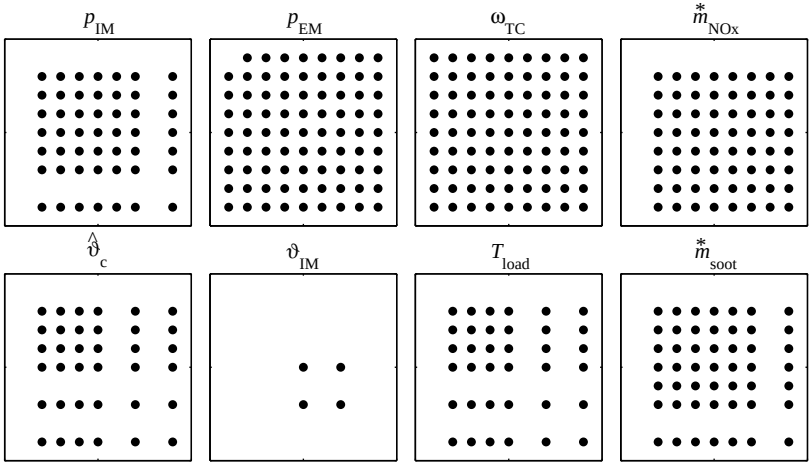

Figure 4.3: Sparsity structure of the Hessian matrices for all state variables and outputs of the engine model, without EGR and without additional restrictions. The fuel consumption is not shown since all entries are zero (it is linear in the control input m_{fcc}).

are able to discover the sparsity pattern of each individual element of a vector-valued function and only propagate the relevant derivatives. If this procedure succeeds, AD should outperform FD, which always evaluates the full model function even if most of its outputs are not required.

4.3 Driving cycles and test cases

Various segments of the World-Harmonized Transient Cycle (WHTC) [81] are used as test cases. This driving profile prescribes the engine speed and the load torque over time.[2] Figure 4.4 displays the full cycle (top), with the individual test cases used in this paper indicated by the shaded areas. The test cycles are shown in the middle and bottom plots, scaled for engine A. Here, the shaded areas indicate drag phases. Test cycles d2, 2 and 4 are contained within cycle 7. The detailed plot of test cycle 8, scaled for engine B, is provided in the left-hand plot in Fig. 5.25 on page 310.

Since the profile demands values for the engine speed and the load torque at each second only, shape-preserving cubic splines are used for the interpolation. Compared to a linear interpolation, this method smoothes the operating-point trajectory, which improves the numerical properties of the resulting OCP.

The shaded areas in the middle and bottom plots of Fig. 4.4 indicate drag phases. During these intervals, the injection is cut off and the engine is motored at the prescribed speed. The individual test cases are characterised in the following list. Cycles d1 and d2 are used to illustrate the inclusion of drag phases in the OCP for diesel engines in Sec. 4.4.

1. Short, simple cycle used for first tests of the algorithms and for illustration purposes.

2. Short segment with a singular arc. The effect of the regularisation is illustrated on this cycle.

3. Longer, realistic driving profile, but without drag phases.

[2] For passenger cars and light commercial vehicles, a trajectory for the vehicle speed is usually prescribed. From this speed profile, an operating-point trajectory for the engine can be calculated by means of a vehicle emulation [147].

4. Longer, realistic driving profile with many drag phases. The presence of motored phases renders the OCP more inhomogeneous and thus more difficult to be solved. Together with cycle 3, this test cycle is used to assess the convergence properties of the various algorithms.

5. Long test cycle which is easy to be solved since almost no transients occur. The cycle is periodic, i.e. it ends at the same operating point as it starts with. By a repetition of the cycle, large-scale OCPs can be constructed that, however, retain the simplicity of the single cycle.

6. Longer, realistic cycle that is used for the experimental validation.

7. Long, realistic cycle that is used for the experimental validation as well as for the identification of a causal control structure in Sec. 5.2.3.

8. A single load step at almost constant engine speed used for the parametric study described in Sec. 5.2.4.

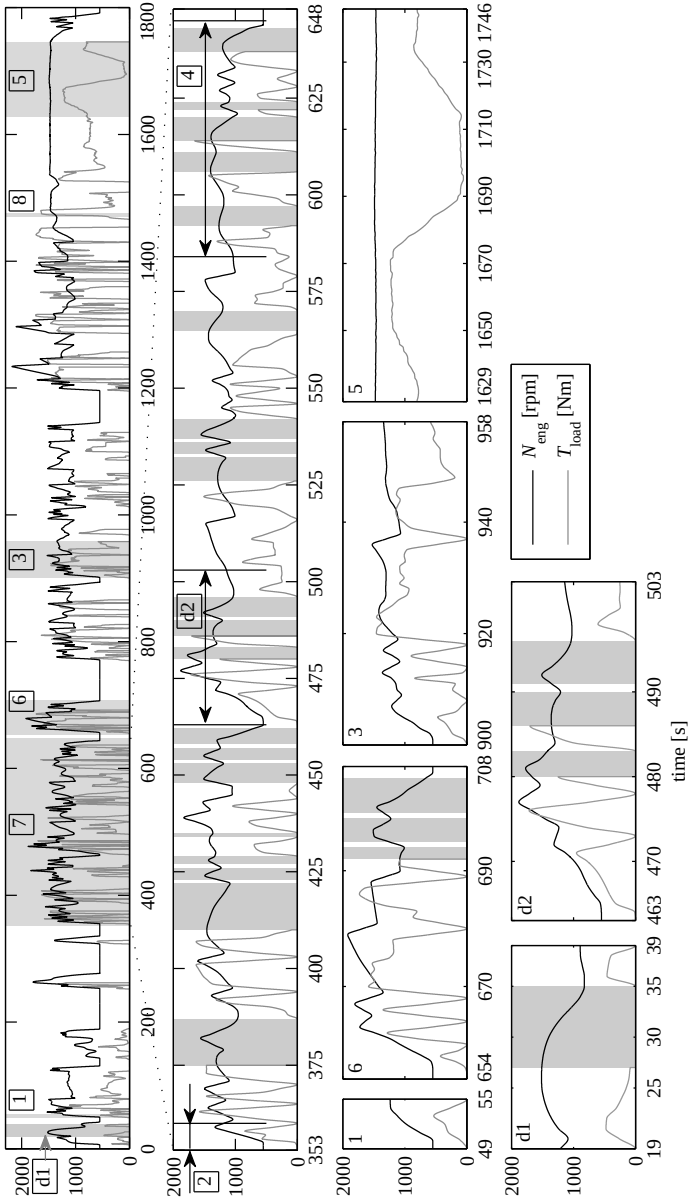

Figure 4.4: The full WHTC and the segments used as test cycles.

4.4 Drag phases

Transient driving cycles usually comprise drag phases. Thus, an optimal-control framework for diesel engines has to be able to handle this operating mode. During drag phases, the fuel injection is cut off, and the engine is motored at the prescribed rotational speed. In a vehicle, this situation occurs, for example, when the driver disengages the accelerator when approaching a traffic junction, or when driving downhill. The structure of the OCP changes in these phases, which has to be carefully accounted for in the numerical framework.

This section shows that the straightforward formulation of drag phases in the OCP needs to be adapted such that the results are plausible considering the driveability. Concurrently, the optimal solution becomes more suitable for the automatic parametrisation or for a meaningful assessment of causal control strategies. Moreover, the formulation of the OCP becomes simpler, since drag phases no longer require any special treatment. As a further side effect, the proposed modifications reduce the search space of the optimisation problem. The resulting faster convergence towards the optimal solution is a welcome fact, especially when dealing with long driving cycles.

To simplify the illustration of the main ideas, first-order collocation on a uniform grid with step size h is assumed throughout this section.

Model causality The driving cycle prescribes trajectories of the engine speed and the load torque. Thus, the momentum balance

$$\Theta_{\mathrm{eng}} \cdot \frac{d\omega_{\mathrm{eng}}}{dt} = T_{\mathrm{eng}}(m_{\mathrm{fcc}}) - T_{\mathrm{fric}} - T_{\mathrm{gex}} - T_{\mathrm{load}} \tag{4.7}$$

has to be honored. Recall that the gas-exchange losses T_{gex} are de-

fined by the pressure difference between the intake and the exhaust manifolds[3],

$$T_{\mathrm{gex}} \approx \frac{V_{\mathrm{d}}}{4\pi} \cdot (p_{\mathrm{EM}} - p_{\mathrm{IM}}). \tag{4.8}$$

For a pressure rise over the engine, the cylinders have to pump their charge from a lower to a higher pressure, absorbing some of the power delivered by the combustion process. The friction torque T_{fric} is modelled as a second-order polynomial in the engine speed.

It would be possible to prescribe both the engine speed ω_{eng} and the load torque T_{load} and use the inverted formulation of the model for the combustion efficiency to obtain the fuel mass m_{fcc}. However, this formulation has been found to result in an NLP which is difficult to solve, and often, numerical problems lead to a premature termination of the optimisation. By adding the fuel mass as a control input and implementing the momentum balance (4.7) as a constraint, the optimisation converges faster and runs more stably. The reason is that by introducing an additional control and a constraint, which may be violated during the optimisation, the NLP solver is granted more degrees of freedom. Thus, it is less prone to get stuck in local minima or to obtain infeasible approximations of the problem during any one iteration.

For these reasons, m_{fcc} is used as an input, and (4.7) is introduced as an additional equality constraint. In the transcribed problem, a

[3] Of course, the actual pressure difference had to be integrated over the gas-exchange cycle. However, $p_{\mathrm{EM}} - p_{\mathrm{IM}}$ often is a close approximation to the mean pressure difference and the only option available in mean-value models [92].

consistent discretisation has to be applied, resulting in

$$\Theta_{\mathrm{eng}} \cdot \frac{\omega_{\mathrm{eng}}(t_{k+1}) - \omega_{\mathrm{eng}}(t_k)}{h} = \left\{T_{\mathrm{eng}}(m_{\mathrm{fcc}}) - T_{\mathrm{fric}} - T_{\mathrm{gex}} - T_{\mathrm{load}}\right\}\Big|_{t=t_{k+1}}. \quad (4.9)$$

4.4.1 Straightforward approach

In driving cycles, the desired engine speed and load torque are specified on a one-second grid. During drag phases, the fuel injection has to be cut off, and the engine is motored along the prescribed engine-speed trajectory. The resulting drag torque, which the testbench brake has to overcome to follow the prescribed speed, is neither limited nor penalised. At the end of a drag phase, the torque is linearly ramped towards the next prescribed value.

These characteristics have to be accounted for by the transcribed OCP. Figure 4.5 illustrates the implementation in the case of first-order Radau collocation, i.e. Euler backward discretisation. More generally, the methodology presented is applicable to all integration schemes that include the right but not the left boundary of the interval, such as Radau collocation. The following steps are necessary.

1. Adapt the discretisation grid to have a grid point at the drag-phase boundaries t_{bm-1} and t_{em}. If the length of the affected intervals is not changed by more than a certain fraction, e.g. 25%, relocate the closest grid point (illustrated at the left boundary in Fig. 4.5). Otherwise, insert an additional grid point (right boundary in Fig. 4.5).

2. If both boundaries of a discretisation interval are inside or on the boundaries of a drag phase, eliminate the engine-speed continuity constraint (4.9) for this interval. The constraints can be actually removed from the NLP or just set to zero.

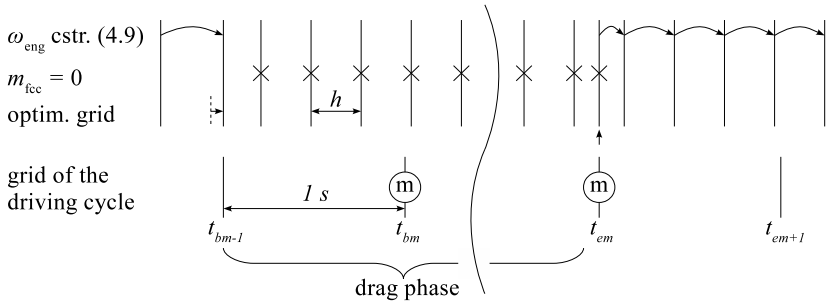

Figure 4.5: Including a drag phase in the transcribed OCP. On top, it is indicated on which intervals the engine-speed continuity constraint (4.9) is imposed and at which discretisation points the fuel injection is cut off. The encircled "m" indicates samples at which the driving cycle prescribes "motored" operation. For illustration, an arbitrary discretisation step-size h is adopted.

3. For all grid points $\in (t_{mb-1}, t_{em}]$, set $m_{\text{fcc}} = 0$. Again, either the lower and upper bounds on the corresponding NLP variables can be set to zero, or the variables may be actually eliminated.

4. Use a forward simulation with a set of reasonable control trajectories to determine a plausible drag torque at t_{em}. Interpolate between this value and the prescribed value at t_{em+1}. Alternatively, only the friction and inertia can be considered to estimate the drag torque at t_{em}. Since friction can be well approximated as a function of the engine speed, no values for the controls have to be estimated.

Note that steps 2 and 3 are simple to implement due to the chosen model formulation described above.

For integration schemes where only the left boundary, both or none of the boundaries of the discretisation interval are included as collocation points, the grid points at which no fuel injection is imposed and

the intervals where the engine-speed constraint is removed have to be adapted accordingly.

Preliminary results The emission limits are chosen as the emissions calculated by a forward simulation of the model using these reference controls. Thus, the optimal solution produces the same cumulated emissions as the reference. Figure 4.6 shows the result of the approach described above, applied to test case d1. The start of injection and the rail pressure are not shown since they are not relevant for the considerations in this section.

The optimal solution has a generally more open VGT, decreasing the pressure rise from intake to exhaust. The fuel savings resulting from these reduced gas-exhange losses and the optimisation of the combustion parameters is 2.00% w.r.t the reference. The massive peak in exhaust pressure during the last few seconds of the drag phase may irritate the observer. In fact, the optimisation fully closes the VGT during this period to increase the boost pressure in the intake manifold by speeding up the turbocharger. The restriction of the exhaust-gas flow, which is dictated by the engine speed and the rising boost pressure, leads to the high pressure in the exhaust manifold.[4] At the end of the drag phase, the VGT is instantly opened, and the exhaust pressure rapidly drops due to the high mass-flow through the turbine. In contrast, the high enthalpy flow temporarily increases the turbocharger speed, which in turn results in a slight rise of the boost pressure. Subsequently, the boost pressure only diminishes slowly due to the turbocharger inertia, causing a positive pressure drop over the cylinders.

[4] At such high inlet pressures, the turbine operates far from its design point and thus its efficiency is low. Therefore, the boost pressure is increased substantially less than the pressure in the exhaust manifold.

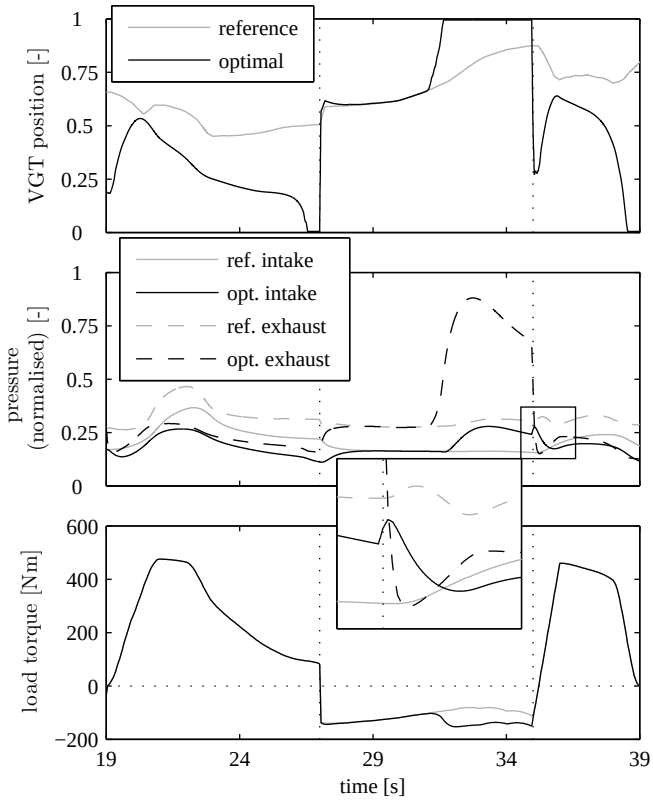

Figure 4.6: Solution of the original OCP compared to the reference, test cycle d1. The vertical dotted lines indicate start and end of a drag phase. The fuel consumption is reduced by 2.00% w.r.t. the reference. For reasons of confidentiality, the pressures are normalised .

During the initial part of the drag phase, the VGT position is not changed at all but remains on the initial trajectory. Actuating the VGT at this time has no effect at all since any changes in the system state are outweighed by the last part of the drag phase. Thus, the VGT position can be chosen arbitrarily. Depending on the strategy implemented in the NLP solver, the initialisation is preserved, or the VGT position may be moved towards the lower or upper boundaries. The SQP method implemented in SNOPT seems to detect early that the VGT position has no influence during this period and thus hardly changes the initial trajectory.

4.4.2 Improved approach

In the formulation developed above, the optimisation exploits the fact that the increased drag torque is not penalised. For comparability to causal control strategies and to obtain a generally more meaningful outcome of the optimisation, the optimal-control problem should be adapted.

A straightforward way of limiting the drag torque would be to fix it at the value obtained when using the reference controls. However, a-priori knowledge of a plausible VGT calibration had to be available to do so. In addition, manual interaction is introduced to the optimisation, which is exactly what is to be eliminated (or at least reduced) by using optimal control to assist the engineer during engine calibration.

A different method of obtaining a reasonable drag-torque reference to use during optimal control is thus proposed. It is based on the fact that, during drag, the only negative torques are caused by friction, the engine inertia and the gas-exchange losses. The former two are largely defined by the driving cycle,

$$T_{\mathrm{cyc}} := T_{\mathrm{fric}} + \Theta_{\mathrm{eng}} \cdot \frac{d\omega_{\mathrm{eng}}}{dt}. \tag{4.10}$$

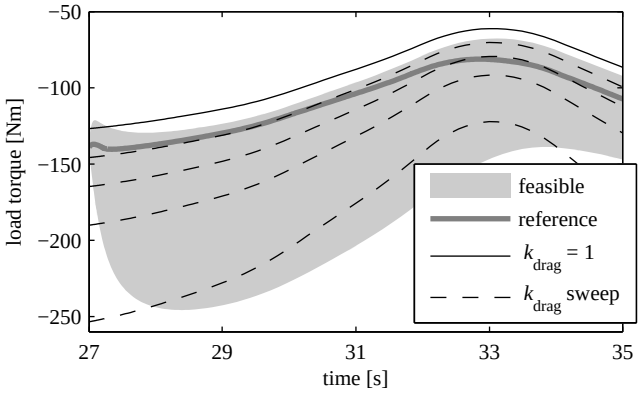

Figure 4.7: Torque during the drag phase of test case d1: feasible range (fully open to closed VGT), result from reference controls, friction and inertia only ($k_{\text{drag}} = 1$), and sweep through $k_{\text{drag}} = \{1.15, 1.3, 1.5, 2\}$. The feasible range starts at a single point since the reference trajectory is applied outside the drag phase.

During the optimisation, the drag torque is limited to a small multiple of this "cycle torque",

$$T_{\text{load}}(t) = k_{\text{drag}} \cdot T_{\text{cyc}}(t), \ \forall \ t \in (t_{bm-1}, t_{em}]. \tag{4.11}$$

To ensure that this requirement does not lead to an infeasible optimal-control problem, the drag torque for fully closed and fully open VGT can be precalculated. Figure 4.7 shows these limits along with the drag-torque references resulting from different values of k_{drag}. A value between 1.15 and 1.3 seems to be a reasonable choice since the resulting drag torque resembles the reference. Note that not allowing any gas-exchange losses ($k_{\text{drag}} = 1$) leads to an infeasible problem.

Torque as a lower inequality bound The drag torque demanded by (4.11) may be below the reachable region in the beginning of the drag

phase, see Fig. 4.7. Even if the VGT is fully closed, less torque is required to motor the engine. Thus, if exactly this drag torque is prescribed, an infeasible optimisation problem results. The solution is to impose the demanded torque as a lower bound over time, replacing (4.7) by

$$T_{\mathrm{eng}}(m_{\mathrm{fcc}}) - T_{\mathrm{fric}} - T_{\mathrm{gex}} - \Theta \cdot \frac{d\omega_{\mathrm{eng}}}{dt} \geq T_{\mathrm{load}}. \tag{4.12}$$

As indicated in (4.9), the first derivative of the prescribed engine speed has to be calculated consistently with the discretisation of the system dynamics.

With this formulation, the structure of the OCP is the same for nominal operation and drag phases. During nominal operation, the optimisation exactly produces the desired load torque, since otherwise, the fuel consumption would increase. During the initial part of a drag phase, the inequality constraint may be inactive.

Arguably, the introduction of many inequality constraints complicates the solution of the resulting NLP. Especially for SQP methods, the need to iteratively approximate the set of inequality constraints which are active at the solution may decrease the performance. However, two facts mitigate this drawback.

- The engine (or inner) torque $T_{\mathrm{eng}}(m_{\mathrm{fcc}})$ is almost proportional to the injected fuel mass, which is a control input. Thus, it is "easy" for the solver to honor the inequality (4.12) by adjusting m_{fcc} accordingly.

- The torque inequality is the only limit on the injection mass m_{fcc}. It directly opposes the objective to reduce the fuel consumption. During each NLP iteration, the constraints are linearised. With the almost-linearity of the torque constraint w.r.t. m_{fcc}, the points at which the constraint is active should be already disclosed dur-

ing the first few iterations of the NLP solver. More importantly, the active set hardly changes during subsequent iterations due to the direct and almost linear relation of the torque constraint and the fuel consumption.

4.4.3 Comparison of the formulations

Figure 4.8 shows the optimal VGT trajectories resulting from the adapted formulation using various values for k_{drag}. A higher value leads to a more closed strategy during drag, followed by a slightly more opened VGT during the first 1-2 seconds after the drag phase. Because the turbocharger speed is higher in this case, slightly more time may pass before the VGT needs to be closed again to extract a sufficient amount of enthalpy from the exhaust gas.

Influence on fuel consumption and solution time. The question left to answer is how the choice of the desired drag torque influences the fuel consumption on the driving cycle. To assess this effect, the more

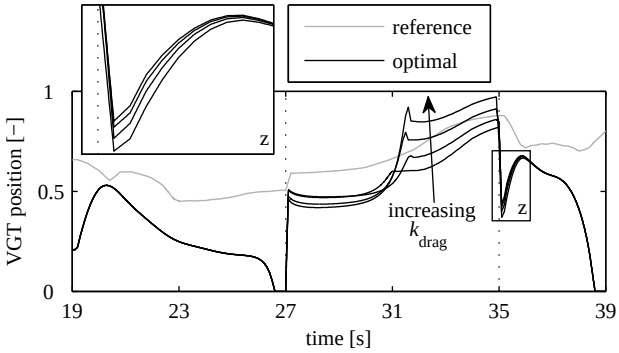

Figure 4.8: VGT position on test case d1, reference and results from optimisation with $k_{\text{drag}} = \{1.15, 1.2, 1.3, 1.5\}$. For the optimisation, the VGT position was initialised at 50% during the drag phase.

Table 4.1: Fuel savings and runtimes of SNOPT for the different ways of handling the drag torque, VGT initialised at 50% during drag phases.

case	*fuel w.r.t. ref.*	*iterations*	*opt. time*
original formulation	−2.44%	201	244.8s
$k_{\text{drag}} = 2$	−2.27%	179	209.7s
$k_{\text{drag}} = 1.5$	−2.13%	113	128.7s
$k_{\text{drag}} = 1.3$	−2.06%	102	126.5s
$k_{\text{drag}} = 1.2$	−2.02%	92	102.1s
$k_{\text{drag}} = 1.15$	−2.00%	68	97.1s

representative test cycle d2 is used. The original formulation is compared to the adapted formulation proposed in Sec. 4.4.2, again using various values for k_{drag}.

Besides the fuel consumption, the complexity of the problem is affected by the choice of the formulation and the parameter k_{drag}. For the original formulation or a large value of k_{drag}, the initial guess is far from the optimal solution, and thus the NLP solver requires more iterations to converge to the optimum.

Figure 4.9 shows the optimal VGT trajectories. The other control inputs are not shown since their optimal trajectories are not affected by the choice of how to deal with drag phases and are not relevant for the considerations here. As for test cycle d1, the different cases only differ during the drag phases and a few seconds afterwards. Table 4.1 lists the fuel savings compared to the reference and the number of iterations and the total time the optimisation requires. As expected, the number of iterations and the overall time for solving the OCP decrease when prescribing a more stringent torque limit.

The fuel consumption is slightly affected by the choice of the drag-torque level. Thus, k_{drag} defines a tradeoff between minimal fuel con-

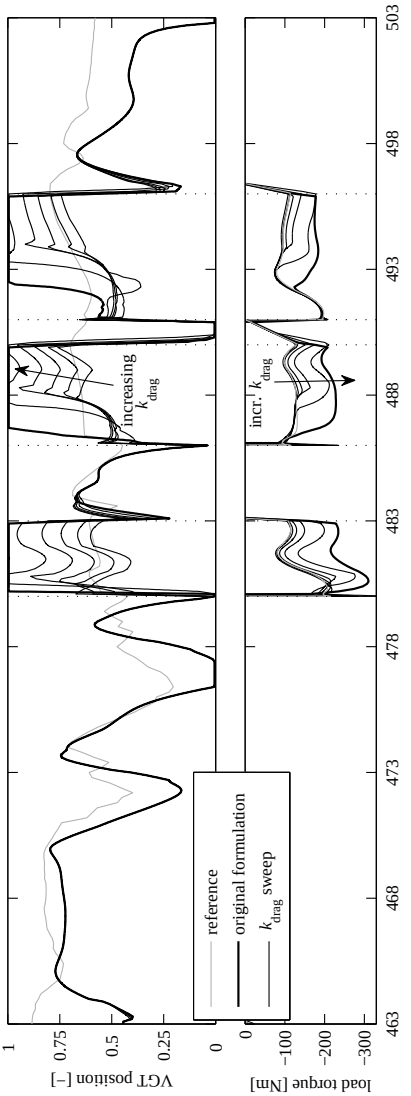

Figure 4.9: Optimal VGT position when not penalising the drag torque (original problem formulation) or limiting it by setting $k_{\text{drag}} = \{1.15, 1.2, 1.3, 1.5, 2\}$. Test cycle d2.

sumption on the driving cycle and the driveability. The term "driveability" is used here to denote how much braking torque the engine produces when coasting. In a vehicle, the engine speed (for a fixed gear) results from the longitudinal dynamics. A high drag torque decelerates the vehicle faster, resulting in shorter coasting phases (e.g. when approaching a junction or a traffic light). Besides being unpleasant, this fast deceleration may require the driver to reengage the throttle pedal, resulting in a higher overall fuel consumption. When the optimal controls are intended to derive causal control strategies, it thus is preferable to limit the drag torque by setting k_{drag} to a reasonably small value.

Since in a causal environment the duration of a drag phase is not known in advance, a general control strategy for coasting operation has to be derived. The maps for this operating mode should be derived from the last part of each drag phase in the OCP. The initial part is not meaningful since the controls have no authority, as described in the context of the preliminary results in Sec. 4.4.1.

4.5 Iterative model refinement

The common approach to dynamic optimisation is to use a large amount of stationary or transient measurement data to identify a globally valid model. Relying on this model, the OCP is solved at once. This approach assumes that all influences on the relevant outputs are captured by the model and the selected set of control inputs. This section proposes an iteration between the dynamic optimisation, restricted to the current model-validity range, and the refinement of the model using specific transient measurements. This approach avoids measurements in irrelevant regions. Furthermore, refining the model using transient measurements on the driving cycle at hand allows to account for the effect of all influences that are missing in the model. This compensation mechanism allows for the use of a simple mean-value model for the air path as well as empirical setpoint-relative emission models. It is thus not required to develop sophisticated combustion models that have to provide reliable far-field extrapolation.

The drawback of the method is that multiple runs of the transient cycle have to be performed during each iteration. The procedure is thus best suited for compact test cycles containing all relevant transient patterns in condensed form. Alternatively, some recurring transient patterns such as load steps or accelerations can be optimised separately to gain insight into the structure of the optimal trajectories of the control inputs.

Similar approaches were already used in the early days of automated engine calibration [52, 66]. However, due to limitations in computer hardware and software, a predefined structure of the control system had to be assumed. Furthermore, the transient measurements were used to directly construct gradient information for the optimisation, since the vast amount of model evaluations required to numerically solve a nonlinear OCP would have been prohibitive at that time. Nowadays, such

computing power is readily available. Thus, a physics-based model for the air-path is used to capture the system dynamics, and time-variable models for the pollutant emissions and for the torque generation are embedded. The accuracy of the air-path model is increased by applying a "generalised Kalman filter". All OCPs occurring within the methodology are solved numerically by the transcription approach described in Chapter 3.

Since a transient model refinement is used, and only relative changes have to be predicted, the air-path model does not have to include the thermal dynamics. Furthermore, the restriction representing the AFM is omitted. To test the procedure outlined here, engine D was used since it was readily available at the institute test-bench and has a flexible test-bench setup.

4.5.1 Time-resolved combustion model

Time-variable quadratic setpoint-deviation models are used for the emissions and for the combustion efficiency. The cross-terms of the full second-order Taylor expansion are omitted since also the $n_\mathrm{u}(n_\mathrm{u}-1)/2$ cross variations would need to be performed to reliably identify the corresponding model coefficients. The model coefficients are identified around the current reference trajectory during each iteration. The vector $\boldsymbol{w} = (p_\mathrm{IM}, x_\mathrm{BG}, \varphi_\mathrm{SOI}, m_\mathrm{fcc})^T$ denotes all inputs to the combustion model. The model for each output thus reads

$$y_i(t) = y_{\mathrm{ref},i}(t) + \boldsymbol{k}_\mathrm{lin}(t)^T \cdot \Delta\boldsymbol{w}(t) + \Delta\boldsymbol{w}(t)^T \cdot \boldsymbol{K}_\mathrm{quad}(t) \cdot \Delta\boldsymbol{w}(t), \quad (4.13a)$$

with

$$\boldsymbol{K}_\mathrm{quad}(t) = \mathrm{diag}\big(k_{\mathrm{quad},1}(t), \ldots, k_{\mathrm{quad},4}(t)\big), \quad (4.13b)$$

$$\Delta\boldsymbol{w}(t) = \boldsymbol{w}(t) - \boldsymbol{w}_\mathrm{ref}(t). \quad (4.13c)$$

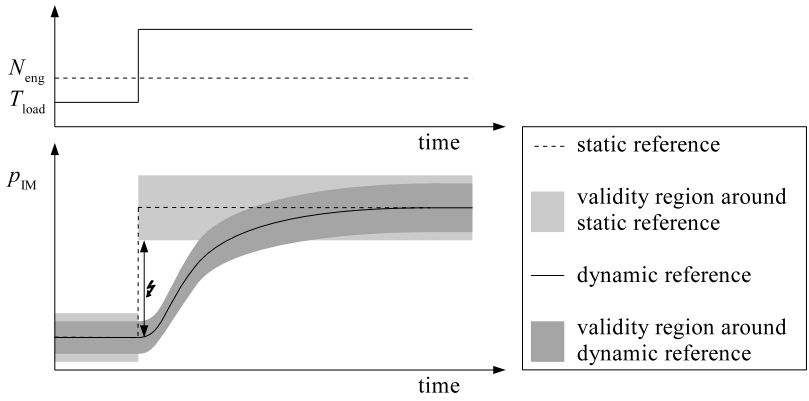

Figure 4.10: Exemplary static and dynamic references over a load step, and the validity regions of the corresponding setpoint-relative models.

Such stepoint-relative models are often used for control and optimisation applications [44, 33]. However, usually the references for the inputs $\boldsymbol{w}$ are stored as static lookup maps over engine speed and load. During transient operation such as a load step, the actual values of the dynamic inputs (p_{IM} and x_{BG} in the case at hand) can be far from the steady-state reference values. By using time-resolved reference values and correction factors, the validity range of the model is relocated to the actually relevant region, which is illustrated in Fig. 4.10.

The outputs of the air-path model, i.e. the dynamic inputs to the combustion model, are denoted by $\boldsymbol{v} := (p_{\mathrm{IM}}, x_{\mathrm{BG}})^T$, with

$$\boldsymbol{v}(t) = \boldsymbol{g}_{\mathrm{AP}}(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \boldsymbol{C}_{\mathrm{AP}} \cdot \boldsymbol{x}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \boldsymbol{x}(t). \quad (4.14)$$

4.5.2 Transient model refinement

This section describes the methods to refine the dynamic part of the model and to identify the combustion model at each iteration. The interaction between the two model portions is indicated at the appropriate locations in the text.

Refinement of the dynamic air-path model A generalisation of a Kalman filter is applied to the dynamic part of the model [109]. In order to correct for systematic model errors, the model equations are augmented by the dynamic and the static corrective variables $\tilde{\boldsymbol{x}}_\mathrm{f}$ and $\tilde{\boldsymbol{x}}_\mathrm{g}$,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{K}_\mathrm{f} \cdot \tilde{\boldsymbol{x}}_\mathrm{f}(t), \tag{4.15a}$$

$$\boldsymbol{x}_\mathrm{corr}(t) = \boldsymbol{x}(t) + \boldsymbol{K}_\mathrm{g} \cdot \tilde{\boldsymbol{x}}_\mathrm{g}(t). \tag{4.15b}$$

The matrices $\boldsymbol{K}_\mathrm{f}$ and $\boldsymbol{K}_\mathrm{g}$ define which dynamics are adjusted and for which state variables the absolute values are corrected.

The optimal-control framework described in Chapter 3 and Appendix D is used to derive the trajectories of the corrective variables. As objective, the integrated squared error in the corrected state variables is used, yielding a least-squares fit of these trajectories. The original state variables remain in the OCP, but the control inputs become time-varying parameters. Instead, the corrective variables $\tilde{\boldsymbol{x}}(t) = (\tilde{\boldsymbol{x}}_\mathrm{f}^T(t), \tilde{\boldsymbol{x}}_\mathrm{g}^T(t))^T$ are optimised in the OCP,

$$\min_{\boldsymbol{x}(t), \tilde{\boldsymbol{x}}(t)} \int_0^{t_\mathrm{f}} \boldsymbol{b}^T \cdot \boldsymbol{K}_\mathrm{g}^T \cdot \Big(\underbrace{\boldsymbol{x}(t) + \boldsymbol{K}_\mathrm{g} \cdot \tilde{\boldsymbol{x}}_\mathrm{g}(t)}_{\boldsymbol{x}_\mathrm{corr}(t)} - \hat{\boldsymbol{x}}(t) \Big)^2 \, dt \tag{4.16a}$$

$$\text{s.t.} \qquad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \hat{\boldsymbol{u}}(t)) + \boldsymbol{K}_\mathrm{f} \cdot \tilde{\boldsymbol{x}}_\mathrm{f}(t). \tag{4.16b}$$

Quantities with a hat, e.g. $\hat{\boldsymbol{x}}$, denote measured signals. The vector $\boldsymbol{b}$ contains the weights to put more emphasis on the accuracy of some of the corrected state variables. The regularisation term introduced in Sec. 3.8 can be used to penalise fast changes of corrective variables. In fact, smooth trajectories are desirable since the model errors are assumed to be of a systematic nature.

If the data from a single measurement is used to identify the corrective variables by solving (4.16), a perfect match of the corrected state trajectories is obtained by adjusting $\tilde{\boldsymbol{x}}_\mathrm{g}$ only. Therefore, multiple measurements need to be considered simultaneously. In the optimal-control framework, N_m instances of the model are stacked to yield a new system with $N_\mathrm{m} \cdot n_\mathrm{x}$ state variables. The error in the relevant outputs is cumulated over all measurements.

$$\min_{\boldsymbol{x}^{(1)}(t),\ldots,\boldsymbol{x}^{N_\mathrm{m}}(t),\tilde{\boldsymbol{x}}(t)} \sum_{k=1}^{N_\mathrm{m}} \int_0^{t_\mathrm{f}} \boldsymbol{b}^T \cdot \boldsymbol{K}_\mathrm{g}^T \cdot \Big(\underbrace{\boldsymbol{x}^{(k)}(t) + \boldsymbol{K}_\mathrm{g} \cdot \tilde{\boldsymbol{x}}_\mathrm{g}(t)}_{\boldsymbol{x}_\mathrm{corr}^{(k)}(t)} - \hat{\boldsymbol{x}}^{(k)}(t) \Big)^2 \, dt \quad \text{(4.17a)}$$

$$\text{s.t.} \quad \begin{pmatrix} \dot{\boldsymbol{x}}^{(1)}(t) \\ \vdots \\ \dot{\boldsymbol{x}}^{(N_\mathrm{m})}(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}(\boldsymbol{x}^{(1)}(t), \hat{\boldsymbol{u}}^{(1)}(t)) + \boldsymbol{K}_\mathrm{f} \cdot \tilde{\boldsymbol{x}}_\mathrm{f}(t) \\ \vdots \\ \boldsymbol{f}(\boldsymbol{x}^{(N_\mathrm{m})}(t), \hat{\boldsymbol{u}}^{(N_\mathrm{m})}(t)) + \boldsymbol{K}_\mathrm{f} \cdot \tilde{\boldsymbol{x}}_\mathrm{f}(t) \end{pmatrix}. \quad \text{(4.17b)}$$

Identification of the combustion model The models for the emissions and the torque generation are re-identified at each iteration. In the final version of the algorithm, all data collected so far should be used during the identification to successively expand the model validity region over the iterations. It is an open point whether the model structure should be adapted, i.e. its complexity increased according to the data available. Alternatively, the measurements could be weighted

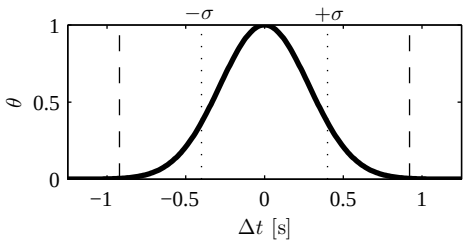

Figure 4.11: Windowed Gauss function used for the time-averaging of the combustion model. The dashed lines delineate the window.

by their distance to the current $\boldsymbol{w}(t)$ to yield a tradeoff between the representation of local and far-field trends. For now, the variations around the initial trajectory are considered only, and the quadratic setpoint-relative model described in Sec. 4.5.1 is used.

A slight averaging w.r.t. time is performed to suppress measurement noise and to provide a smooth model, which is advantageous in the context of optimisation. A windowed Gauss curve is used to weight the preceding and consecutive samples for the identification of the model at each time instant,

$$\theta(\Delta t) = \begin{cases} \exp\left(-\frac{\Delta t^2}{\sigma^2}\right), & \text{if } |\Delta t| < \sqrt{-\sigma^2 \cdot \log_e(\theta_{\text{cut}})}, \\ 0, & \text{else.} \end{cases} \tag{4.18}$$

The parameter θ_{cut} defines the value for θ at which the Gauss curve is cut. Figure 4.11 shows the curve for the parameter values $\sigma = 0.4s$ and $\theta_{\text{cut}} = 0.5\%$, which are found to be a reasonable choice.

For each output y_i, a weighted linear least-squares regression is used to identify the coefficients at each sampling point t_k. The $N_{\text{m}} - 1$ variations are used, which are again denoted by the superscript index

in round brackets. The equation system

$$(\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X}) \cdot \boldsymbol{p} = \boldsymbol{X}^T \boldsymbol{W} \Delta \boldsymbol{y}_i, \tag{4.19a}$$

is solved for $\boldsymbol{p}$, where

$$\boldsymbol{p} = \left(k_{\text{lin},1}(t_k), \ldots, k_{\text{lin},4}(t_k), k_{\text{quad},1}(t_k), \ldots, k_{\text{quad},4}(t_k)\right)^T, \tag{4.19b}$$

$$\Delta \boldsymbol{y}_i = \begin{pmatrix} \boldsymbol{y}_i(t_{-l}) - \boldsymbol{y}_{\text{ref},i}(t_{-l}) \\ \vdots \\ \boldsymbol{y}_i(t_l) - \boldsymbol{y}_{\text{ref},i}(t_l) \end{pmatrix}, \tag{4.19c}$$

$$\boldsymbol{X} = \begin{bmatrix} \Delta \boldsymbol{w}_1(t_{-l}) & \cdots & \Delta \boldsymbol{w}_4(t_{-l}) & \Delta \boldsymbol{w}_1(t_{-l})^2 & \cdots & \Delta \boldsymbol{w}_4(t_{-l})^2 \\ \vdots & & \vdots & \vdots & & \vdots \\ \Delta \boldsymbol{w}_1(t_l) & \cdots & \Delta \boldsymbol{w}_4(t_l) & \Delta \boldsymbol{w}_1(t_l)^2 & \cdots & \Delta \boldsymbol{w}_4(t_l)^2 \end{bmatrix}, \tag{4.19d}$$

$$\boldsymbol{W} = \text{diag}\big(\theta(t_{-l} - t_k) \cdot \boldsymbol{I}_{N_m - 1}, \ldots, \theta(t_l - t_k) \cdot \boldsymbol{I}_{N_m - 1}\big). \tag{4.19e}$$

Here, l is the number of samples inside the window in both directions, and $\boldsymbol{I}_{N_\text{m}-1}$ is the identity matrix. Each vector $\Delta \boldsymbol{w}_j := \left(w_j^{(1)} - w_{\text{ref},j}, \ldots, w_j^{(N_\text{m}-1)} - w_{\text{ref},j}\right)^T$ stacks all variations (similarly for $\boldsymbol{y}_i$). Here, $\Delta \boldsymbol{w}_j^2$ denotes element-wise squaring.

4.5.3 Iterative procedure

The inputs to the procedure are a transient driving profile, consisting of engine speed and load torque trajectories, and any calibration of the ECU that is able to operate the engine along this profile. The iterative procedure relies on two modes of operating the engine on the test-bench.

A) The ECU with its standard calibration controls the engine. The testbench controller is used to follow the desired profiles of the engine speed (by controlling the brake torque) and the load torque (by controlling the fuelling). This mode is used for the initialisation of the iterative procedure. The resulting trajectories of the controls, including the injected fuel mass, are recorded.

B) Time-resolved trajectories are prescribed for all control inputs using a rapid-prototyping module and bypasses of the ECU. The testbench controller is only used to follow the engine-speed profile. This mode is used for validation runs as well as for all variations.

The individual steps of the iterative procedure are described in the following list.

1. Initialisation: drive the profile in mode A. Save the resulting trajectories of the control inputs, set them as the "current controls" $\boldsymbol{u}(t)$. Set the corrective variables $\tilde{\boldsymbol{x}}(t)$ to constant zero.

2. Perform $1 + 2 \cdot n_{\mathrm{u}}$ testbench runs in mode B. Thereby, apply

   (a) the current controls $\boldsymbol{u}(t)$, and

   (b) isolated perturbations of all controls in both directions, i.e. $u_i(t) \leftarrow u_i(t) \pm \Delta u_i$, for $i = 1, \ldots, n_{\mathrm{u}}$.

3. Use the measurement data from step 2 to identify the corrective variables $\tilde{\boldsymbol{x}}(t)$ by solving (4.17). Run simulations of the refined air-path model for all variations, and save the resulting corrected state trajectories $\boldsymbol{x}_{\mathrm{corr}}(t)$.

4. Use the state trajectories of the air-path model from step 3 to identify the torque and emission models around the current references by solving (4.19).

5. Solve the control and state constrained OCP (3.23) to derive the improved control trajectories $\boldsymbol{u}^*(t)$. Thereby, set

$$\underline{\boldsymbol{u}}(t) = \boldsymbol{u}(t) - k_\mathrm{u} \cdot \Delta\boldsymbol{u}, \quad \overline{\boldsymbol{u}}(t) = \boldsymbol{u}(t) + k_\mathrm{u} \cdot \Delta\boldsymbol{u}, \tag{4.20a}$$

$$\underline{\boldsymbol{x}}(t) = k_\mathrm{x} \cdot \min\{\boldsymbol{x}(t)^{(k)} |\ k = 1, \ldots, 1 + 2 \cdot n_\mathrm{u}\}, \tag{4.20b}$$

$$\overline{\boldsymbol{x}}(t) = k_\mathrm{x} \cdot \max\{\boldsymbol{x}(t)^{(k)} |\ k = 1, \ldots, 1 + 2 \cdot n_\mathrm{u}\}. \tag{4.20c}$$

6. Set $\boldsymbol{u}(t) \leftarrow \boldsymbol{u}^*(t)$, repeat steps 2.-5. until the change in the controls is small.

Using the simulated state trajectories during the identification of the combustion model in step 4 ensures a consistent prediction of the emissions and of the torque inside the validity region. This fact is important since also the optimisation in step 5 is restricted to this region. This "trust region" can be slightly expanded by the factors k_u and k_x.

Since the identification of the models for the air path and for the combustion are identified separately, the physical causality is preserved. More exactly, there is no way that the combustion model corrects errors in the air-path model or vice-versa. Although a combined identification could yield a slightly higher accuracy, it would introduce unphysical cross corrections. Furthermore, the identification procedure would become more complex and non-convex.

The results concerning the transient model refinement and the time-variable combustion model are presented in Sec. 5.2.1 below. Preliminary experimental results, as well as the conclusions that can be drawn from them are provided in Sec. 5.3.2.

5 Results

The results are subdivided into three sections. The first one analyses crucial numerical aspects and states the conclusions that can be derived from the results of these tests. The second part covers the engineering aspect by presenting ways of utilising the results obtained with optimal control methods. The last section presents the experimental validation of those results to demonstrate the meaningfulness of the applications described in the first two sections.

5.1 Numerical aspects

First, the results from all tests performed are presented and discussed. Based on these data, conclusions are drawn on what discretisation approach is suitable for the problem at hand, and which numerical methods are preferable for solving the resulting NLPs.

5.1.1 Mesh refinement, regularisation, and MSIM

Small but illustrative examples are presented to analyse the effect of these extensions of the standard transcription method. Where appropriate, the implications for realistically sized problems are indicated in the text.

Mesh refinement Figure 5.1 illustrates the effect of the mesh refinement on test cycle 1. First-order collocation is used, and a relative tolerance of $5 \cdot 10^{-3}$ is requested for the ODE solution. This accuracy is achieved by a uniform discretisation with a step size of 0.1 s. The exact-Newton method of WORHP requires 21 iterations and 13.8 s for the solution of the resulting NLP.

When mesh refinement is applied, an initial step size of 0.25 s is used. Two refinements are performed. The three solution runs of the coarse initial problem and the two refined problems require 17, 6, and 5 iterations, respectively, and a total of 11.0 s. After the second refinement, the desired accuracy is achieved, and the problem is discretised into 45 intervals only, as compared to the 60 intervals resulting from the uniform discretisation.

When this procedure is applied to test cycle 7, the solution time is reduced from 2609 to 1015 s, i.e. from around 43 to 17 minutes. IPOPT requires 3234 s to solve the uniformly discretised test cycle 7. However, in contrast to the SQP method implemented in WORHP, IPOPT is not able to exploit the good initialisation of the refined problems. In fact, the

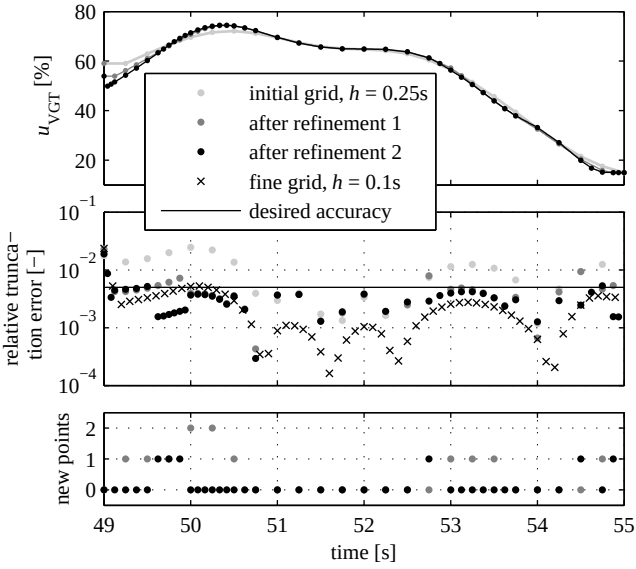

Figure 5.1: Mesh refinement on test cycle 1. First-order collocation is used, and the initial step size is 0.25 s. Two refinements are performed with $k = 0.5$ and $k = 1$, respectively.

solution of the initial and the first refined problems require the same amount of time as the one-off solution of the uniformly discretised problem.

Regularisation Figure 5.2 shows the optimal trajectories of the control inputs when second-order collocation on a uniform grid with a step size of 0.2 s is applied to test cycle 2. As expected, oscillations occur mainly in the region where the lower trajectory constraint for the rail pressure is active. The fuel consumption and the pollutant emissions predicted by the optimisation are verified by an accurate forward simulation, which yields the same results. For a finer discretisation, faster

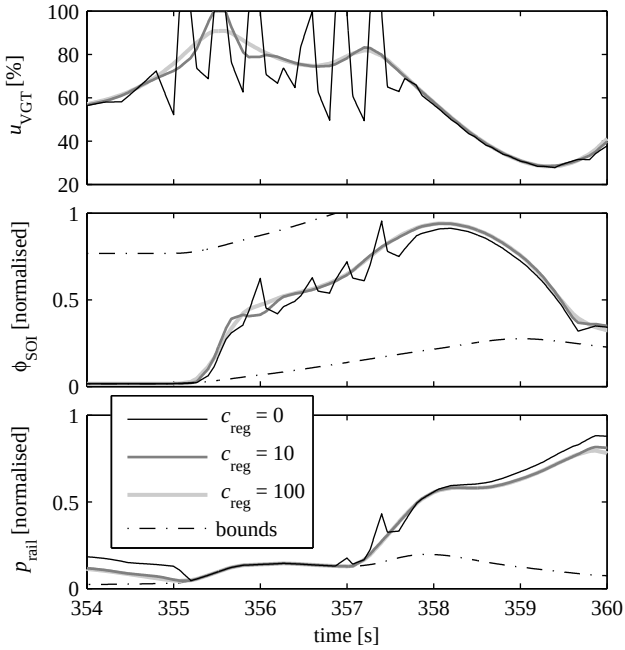

Figure 5.2: Regularisation of the control inputs, test cycle 2. The fuel consumption is reduced by 2.676% by the non-regularised solution and by 2.567% when a regularisation with $c_{reg} = 100$ is applied. Normalised signals are shown for reasons of confidentiality.

oscillations result. Also when only the state variables are resolved more accurately, the oscillations persist.

Consequently, an oscillatory solution actually is optimal. Possibly, the gas-exchange losses are slightly reduced by the oscillations of the pressure in the exhaust manifold induced by the oscillations of the VGT. At the same time, the intake pressure is hardly affected due to the slow dynamics of the turbocharger. Therefore, the air mass-flow remains the same and the soot emissions do not increase.

Since a fast oscillating actuation of the mechanical actuators is not sustainable, regularisation has to be applied. As Fig. 5.2 shows, the regularisation does not change the general shape of the solution. In particular, the solution is identical in non-singular regions. Furthermore, the loss in fuel efficiency when requiring a smooth solution is negligible.

Multiple-shooting imitation mode To further support the proposition that the oscillations are not caused by an inaccurate integration or too fine a resolution of the control inputs, the multiple-shooting imitation mode (MSIM) described in Sec. 3.9 is applied. Figure 5.3 shows the same scenario as that presented in Fig. 5.2 for a piecewise-constant control discretisation of 0.1 s, but using second-order collocation on these intervals. Note that if first-order collocation is applied on the same grid, no oscillations occur at all. Therefore, only a more accurate approximation of the continuous-time problem reveals the singular nature of the problem as indicated in Sec. 3.8.

Figure 5.4 displays the optimal VGT position and the ODE truncation error for the same scenario when piecewise-linear control and third-order collocation is used on a 0.2 s grid. Already for the non-regularised solution, the truncation error is below $3 \cdot 10^{-3}$ for all collocation intervals. The slight reduction of this error when regularisation is applied is a direct consequence of the smoother excitation of the system.

5.1.2 Convergence

Table 5.1 summarises the performance of all NLP solvers described in Sec. 3.2.3 on the short, simple test cycle 1. Several discretisation approaches are tested, from first-order collocation to the pseudospectral method. The regularisation parameter c_{reg} is chosen for each case in-

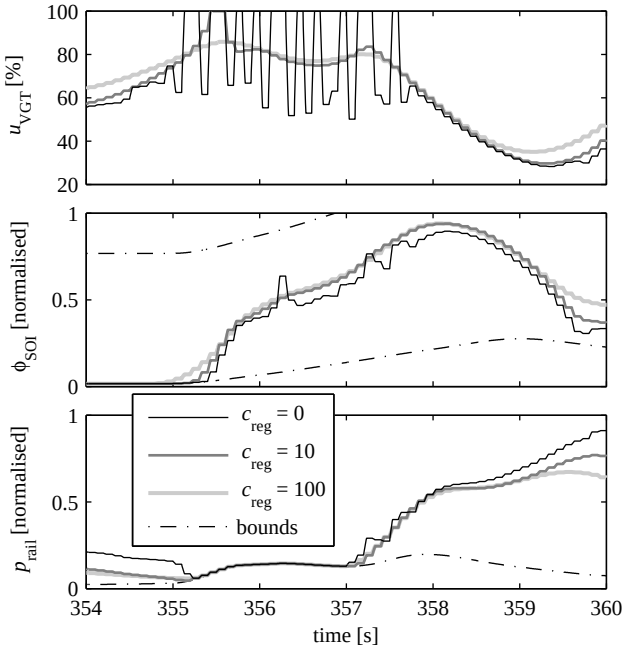

Figure 5.3: Regularisation of the control inputs, test cycle 2. MSIM with piecewise-constant control and second-order collocation on a 0.1s grid is used. The fuel consumption is reduced by 2.844% by the non-regularised solution, and by 2.510% when a regularisation with $c_{\text{reg}} = 100$ is applied. Normalised signals are shown for reasons of confidentiality.

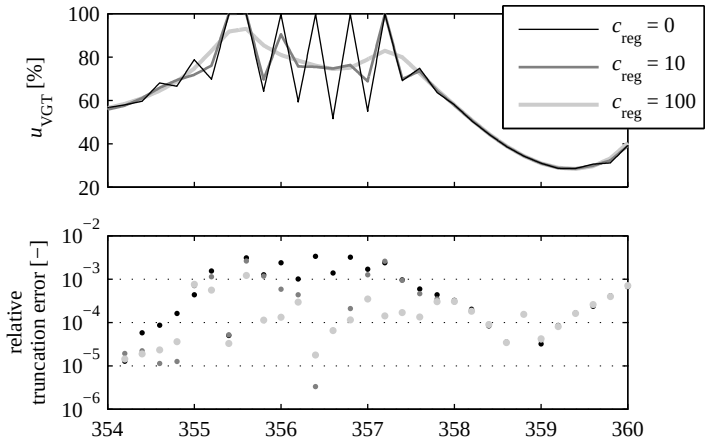

Figure 5.4: Regularisation of the control inputs (only VGT shown since it exhibits the largest oscillations), and truncation error of the ODE solution, test cycle 2. MSIM with piecewise-linear control and third-order collocation on a 0.2s grid is used. The fuel consumption is reduced by 2.677% by the non-regularised solution, and by 2.208% when a regularisation with $c_{\text{reg}} = 100$ is applied. Normalised signals are shown for reasons of confidentiality.

dividually such that a smooth solution results. For the solver WORHP, only the exact Newton method is applied. None of the partitioned BFGS updates yields satisfactory results for the problem at hand.

The same data is provided for the more realistic test cycles 3 and 4 in Table 5.2. None of the methods implemented in KNITRO were able to solve any of the problem instances to the required tolerance within 200 outer iterations. Therefore, this data is not shown. Similarly, IPOPT is most efficient w.r.t. the number of iterations and the solution time when the exact second derivatives are provided. The QN variant did not converge within 200 iterations for any of the tests on cycle 4.

The first and second partial derivatives of the model functions, from which the KKT matrix of the NLP is constructed, are calculated

Table 5.1: Performance of the NLP solvers on test cycle 1, number of outer iterations and overall time required for the solution (in brackets). For the discretisation with order 5^*, piecewise-constant control inputs are imposed by MSIM. The last row is the pure pseudospectral method. Symbols used: h (length of collocation intervals), s (collocation order), n_{NLP} (number of NLP variables), n_{DOF} (degrees of freedom in the NLP), QN/EN (quasi/exact Newton method). For WORHP, only the exact Newton method is shown. For the IP method using a direct solver in KNITRO (KN-IPDIR), only the QN method could solve the problem, which is shown here. The IP-CG method always performed worse and thus is not shown. An accuracy of 10^{-6} is requested w.r.t. optimality and feasibility. If this accuracy was not achieved within 200 iterations, the optimisation was terminated, which is indicated by italic script. Bold script indicates the fastest solution for each case.

h	s	c_{reg}	n_{NLP}	n_{DOF}	*SNOPT*		*IPOPT, QN*		*IPOPT, EN*		*WORHP*	
0.25	1	0	225	61	**20**	**(3.5)**	39	(6.4)	13	(4.5)	18	(6.7)
0.25	3	10	657	190	69	(15.2)	40	(9.9)	**15**	**(9.8)**	18	(11.9)
0.25	5^*	0	1089	60	**23**	**(7.1)**	30	(10.7)	15	(14.3)	21	(23.2)
3.00	15	1	279	78	42	(6.8)	62	(10.2)	**12**	**(4.8)**	21	(8.5)
6.00	30	0	279	80	154	(27.0)	82	(21.9)	**28**	**(10.7)**	27	(13.7)
					KN-SQP, QN		*KN-SQP, EN*		*KN-IPDIR*			
0.25	1	0	225	61	66	(11.3)	58	(29.6)	54	(7.8)		
0.25	3	10	657	190	151	(49.0)	65	(64.9)	70	(19.3)		
0.25	5^*	0	1089	60	136	(82.2)	112	(167.5)	137	(86.8)		
3.00	15	1	279	78	*200*	*(43.8)*	*200*	*(107.1)*	144	(32.3)		
6.00	30	0	279	80	*200*	*(45.9)*	*200*	*(118.2)*	*200*	*(45.9)*		

Table 5.2: Performance of the NLP solvers on test cycles 3 (top) and 4 (bottom). Only the exact Newton methods are shown for IPOPT and WORHP. The regularisation is chosen individually for all discretisation variants such that smooth control-input trajectories are obtained.

h / s	n_{NLP}	*SNOPT*		*IPOPT*		*WORHP*	
0.25 / 1	2097	35	(29.9)	**16**	**(26.4)**	24	(54.5)
0.40 / 3	3924	37	(114.8)	**19**	**(61.1)**	24	(108.1)
0.25 / 5*	10449	**24**	**(114.5)**	17	(153.6)	33	(338.9)
3.00 / 15	2574	51	(120.8)	**18**	**(39.0)**	28	(69.7)
9.60 / 47	2547	179	(554.6)	*200*	*(544.4)*	**26**	**(92.7)**
58.0 / 282	2547	*200*	*(5495.5)*	26	(448.2)	**24**	**(433.7)**
0.25 / 1	2205	**45**	**(29.7)**	28	(46.7)	37	(117.4)
0.40 / 3	4113	114	(311.5)	**29**	**(89.0)**	49	(196.5)
0.25 / 5*	10989	**66**	**(386.1)**	76	(680.2)	33	(405.8)
3.00 / 15	2709	197	(331.8)	*25*[a]	*(65.2)*	**41**	**(104.5)**
10.1 / 50	2709	*200*	*(659.9)*	**36**	**(92.3)**	58	(204.2)
61.0 / 300	2709	*200*	*(6219.3)*	*30*[a]	*(2493.7)*	**30**	**(781.6)**

[a] A segmentation error crashed the optimisation at that iteration.

by forward finite differences (FFD). The convergence behaviour of the NLP solvers is not improved when more accurate derivatives are calculated by algorithmic differentiation (AD). Even the exact Newton methods require the same number of iterations when using AD instead of FFD. Therefore, AD is advantageous only if an implementation is available that is faster than FFD. All implementations of AD for MATLAB tested, namely ADiMat [41], ADMAT 2.0 (Cayuga Research, Waterloo, ON, Canada), INTLAB V6 [160], and the open-source implementation [74] are found to be at least a factor of 1.6 slower than FFD when the first and second partial derivatives of the model functions are calculated.

5.1.3 Partitioned quasi-Newton update

IPOPT is used to generate preliminary results concerning the partitioned quasi-Newton (QN) update for NLPs resulting from the trancription of OCPs by direct collocation. Radau collocation is used, and either the exact Hessian, calculated using central finite differences on the model functions, or a full or the partitioned QN update is supplied to the solver as user-defined Hessian. In all cases, the first derivatives are calculated by forward finite differences.

As a small problem, test case 1 is used, discretised by first-order collocation with a uniform step length of 0.5 s. This discretisation results in $M = 13$ total collocation points and 117 NLP variables. As a more representative problem, test case 3 is considered, and third-order collocation with a step size of 0.8 s is applied, resulting in $M = 217$ collocation points and 1953 NLP variables.

The performance is assessed in terms of the number of iterations required to achieve the requested tolerance. Figure 5.5 shows the convergence behaviour of the three methods for test case 1. The damped BFGS update (3.14) is used. Starting with iteration 5, the full Newton step is always accepted. Thus, the difference between the local quadratic convergence of the exact Newton method and the local superlinear convergence of the full BFGS update becomes obvious. The partitioned update performs substantially better than the full update. When the SR1 update (3.15) is used, 44 or 46 iterations are required for the partitioned and full updates, respectively.[1]

On test case 3, the exact Newton method converges after 16 iterations, and the partitioned BFGS update requires 44 iterations. In con-

[1] The worse performance of the SR1 update, which does not preserve positive definiteness, can be attributed to the fact that IPOPT uses a line-search globalisation. In a trust-region framework, the SR1 update could be advantageous since it can approximate the generally indefinite Hessian of the (full or element) Lagrangian more accurately.

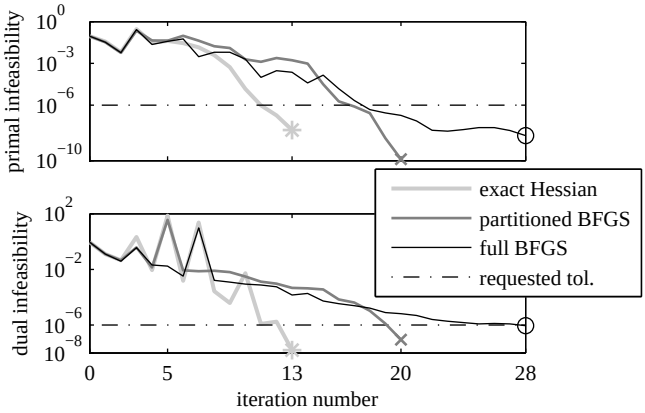

Figure 5.5: Convergence in the primal and the dual variables. Exact Hessian as well as partitioned and full (damped) BFGS update, test case 1.

trast, 157 iterations are necessary when approximating the full Hessian by the BFGS update. The convergence behaviour is illustrated in Fig. 5.6.

The Hessian approximation is initialised by a multiple of the identity matrix in all cases, i.e. $\boldsymbol{B}_0 = \beta \boldsymbol{I}$. A factor of $\beta = 0.05$ is found to be a good choice for the problem at hand. Table 5.3 shows the number of iterations required as the factor for the initial approximation is changed. The partitioned update is quite robust against a bad initial guess, whereas the full update requires a significant number of iterations to recover. This finding confirms the expectation that an accurate approximation is obtained in fewer iterations when the QN update is applied to the small diagonal blocks.

The full BFGS update produces a completely full Hessian approximation after the first update already. Thus, the linear solver requires about 3.35 s for each NLP iteration for test case (b). In contrast, when the correct sparsity pattern can be exploited, which is the case for the

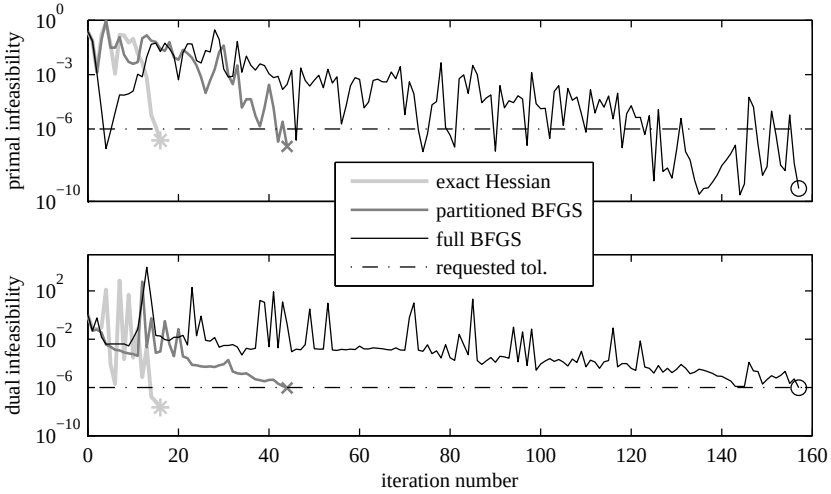

Figure 5.6: Convergence in the primal and the dual variables. Exact Hessian as well as partitioned and full (damped) BFGS update, test case 3.

exact Hessian and the partitioned approximation, one NLP iteration takes just 0.03 s.

Accuracy of the derivatives On various test scenarios, the dependency of the convergence behaviour of the NLP solvers on the accuracy of the derivatives is analysed. Namely, forward FD, centered FD, and AD are applied to the model functions to calculate the first and second

Table 5.3: Effect of the initialisation of the Hessian approximation on the number of NLP iterations required to achieve convergence.

method	$\beta = 0.01$	$\beta = 0.05$	$\beta = 0.1$
full BFGS	36	28	38
partitioned BFGS	22	20	21

partial derivatives. For QN methods, the accuracy of the derivatives evidently is not an imperative requirement. The second-order information is extracted from sampling the first-order derivatives, which can be provided with a sufficient accuracy already by forward FD. This finding is confirmed by practical tests using SNOPT and the QN version of IPOPT.

The situation is more delicate when exact Newton methods are considered. Since the true second-order information of the problem is exploited, the accuracy of this data ought to be important. However, experimental tests with the exact-Newton methods in IPOPT and WORHP reveal that the convergence behaviour of these algorithms is hardly affected at all by the accuracy of the derivatives. Figure 5.7 illustrates this finding on the larger test problem introduced above (test cycle 3).

Arguably, the globalisation strategy plays a dominant role during the first iterations. However, the full Newton step is always accepted starting at iteration 5 for IPOPT, and at iteration 6 for WORHP. The convergence behaviour of a Newton-type solver thus is primarily defined not by the actual accuracy, but rather by the nature of the second derivatives, i.e. whether exact or approximated information is provided. Only if a problem needs to be solved to very tight tolerances, switching to CD or even AD could be advantageous. Furthermore, if the application of AD is possible and an efficient implementation is available that outperforms forward FD, there is no reason for not applying AD.

5.1.4 Computational performance

A qualitative assessment of the computational performance of the NLP solvers is presented in Figs. 5.8 and 5.9. A medium-sized problem is used as test scenario, and all four cores of the processing unit are used to perform the model evaluations in parallel.

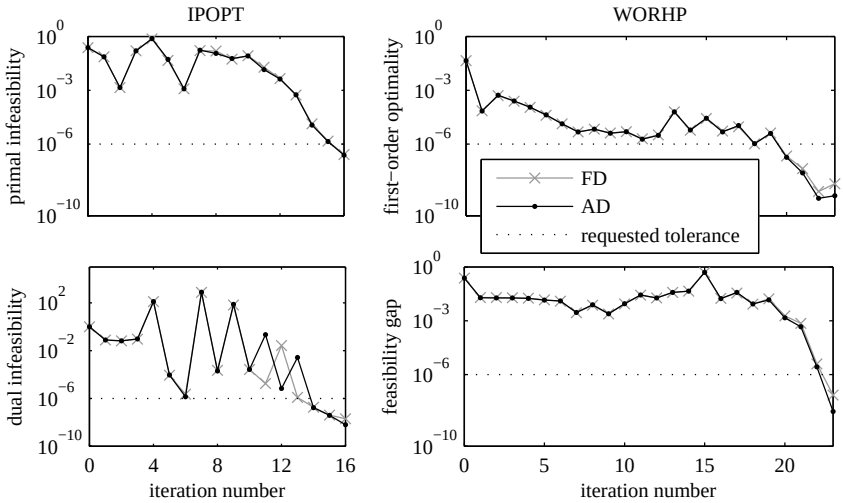

Figure 5.7: Effect of the accuracy of the derivatives on the convergence behaviour of the exact Newton methods in IPOPT and WORHP. Only during the last few iterations, a slightly better performance is observable when AD is used.

The matrix refactorisations within SNOPT clearly outweigh the gain of no Hessian evaluations. IPOPT needs to perform the preconditioning of the sparse KKT system (fill-reducing reordering and symbolic LU factorisation) only once since the problem structure is invariant within an IP method. Whenever the active set of the problem changes substantially, the IP solver for the QP subproblems in WORHP requires a substantial amount of time to solve the QP, which is visible at iterations 2 and 5 (exact Hessian). A refactorisation of the KKT system seems to be performed at those iterations.

The IP solvers implemented in KNITRO exhibit a performance similar to that of IPOPT but require slightly more time since two Jacobian evaluations are performed during each iteration. The CG version is

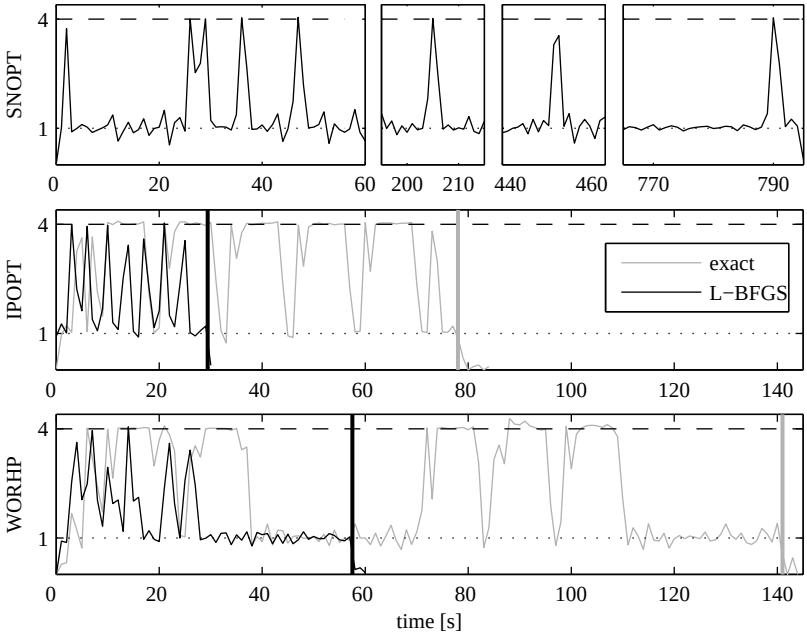

Figure 5.8: Processor utilisation of the NLP solvers over the first five outer iterations. Test case 3, discretised on a uniform grid of 0.1s using third-order Radau collocation, resulting in 15'660 NLP variables. The two horizontal lines denote the load of one or four processor cores, respectively. During evaluation of the Jacobian (narrow peaks) and the Hessian of the Lagrangian (broad peaks), all cores are fully utilised, even if in some instances it does not show clearly due to the 1s grid of the monitoring tool.

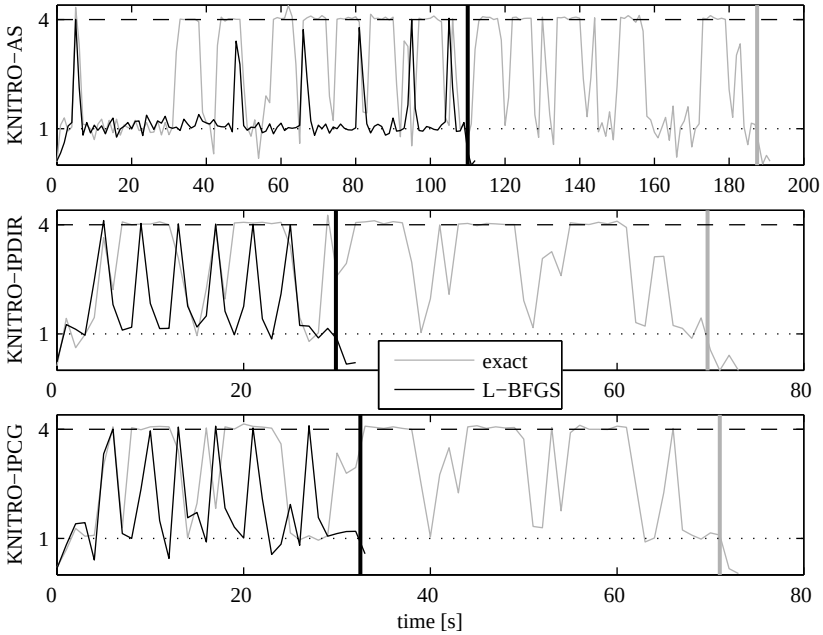

Figure 5.9: Processor utilisation of the three NLP solvers implemented in KNITRO, same scenario as in Fig. 5.8.

slightly slower than the implementation using a direct linear solver. The active-set (AS) solver requires more Jacobian and Hessian evaluations per iteration, and it requires some time to set up the problem.

Large-scale performance Test cycle 5 is repeated multiple times to construct a problem of increasing size that retains the same complexity. Therefore, the performance of the NLP solvers when applied to large problems can be assessed independently of their general convergence behaviour. First-order collocation on a uniform grid with a step size of 0.25 s is used, and the cycle is repeated 1 to 6 times. NLPs with around 4,000 to 25,000 variables result.

The main observations are illustrated in Fig. 5.10. All solvers require a similar number of iterations to solve the differently sized problems. Even the QN methods perform well for the large-scale problems. Therefore, their poor performance on the more difficult test cases, see Sec. 5.1.2, seems not to be a problem of the number of variables, but of the complexity and the nonlinearity of the problem to be solved.

The number of model evaluations per iteration is proportional to the number of discretisation points and thus to the size of the NLP. Therefore, the time required to solve the KKT system or to perform updates of a decomposition directly defines the computational performance of the solvers for large-scale problems. The time required for the (re-)factorisations and for the dense algebra on the reduced problem in SNOPT grows with a power of about 2.5 w.r.t. the problem size as illustrated in Fig. 5.11. All other solvers work on the full but sparse KKT system and exhibit an approximately linear runtime. The proportionality factor between runtime and problem size differs by a factor of 3 from the QN method of IPOPT to the exact Newton method in WORHP. In WORHP, the linear solver has to recalculate or update the indefinite factorisation of the KKT system at the beginning of each outer iteration.

For the full-matrix approaches just described, the fraction of the overall solution time required for the model evaluations, which are performed in parallel on four cores, is between 40% (KNITRO) and 70% (IPOPT). For SNOPT, this fraction is below 1% for large problems.

5.1.5 Large-scale algorithm or time subdivision?

First, the time decoupling inherent to the OCP of diesel engines is illustrated. The results from the time subdivision, combined with parallel computing and an iterative mesh refinement, then are presented for test cycles 3 and 7. SNOPT is used to solve the subproblems of the subdivided problem since they are relatively small, and an effective

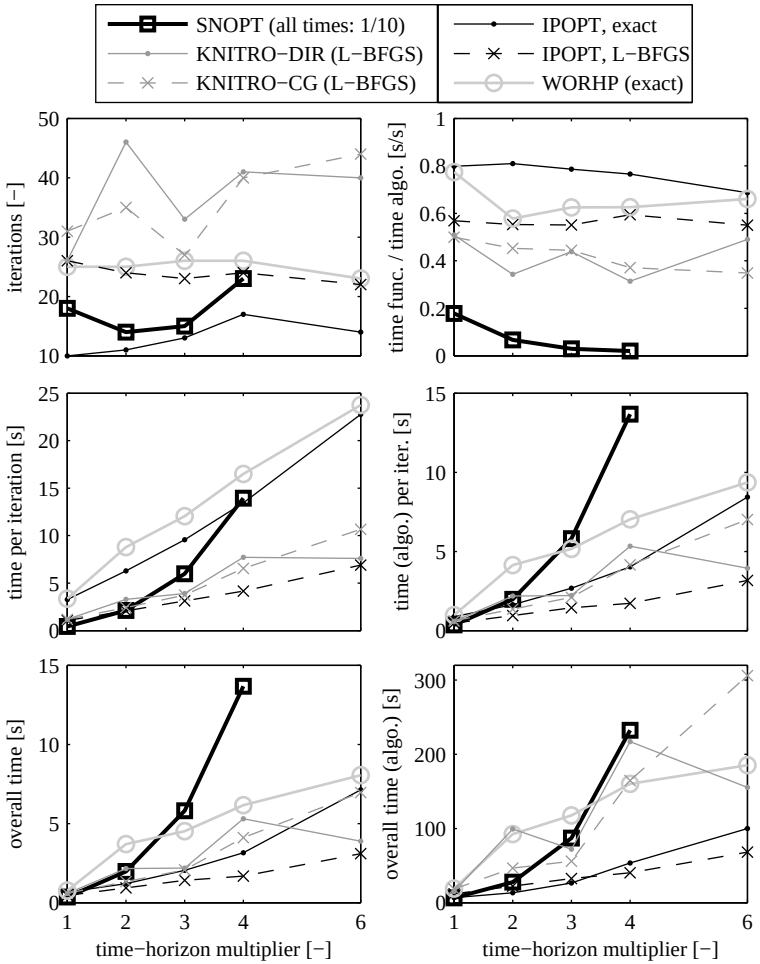

Figure 5.10: Performance of the NLP solvers for an increasing problem size. Multiple instances of test case 5 are appended to enlarge the time horizon without changing the problem structure. First-order collocation on a 0.25 s grid is used, resulting in 4,221 NLP variables for the original test case. When the test case is repeated 6 times, 25,281 NLP variables result. For SNOPT, all runtimes are divided by 10 in the plots.

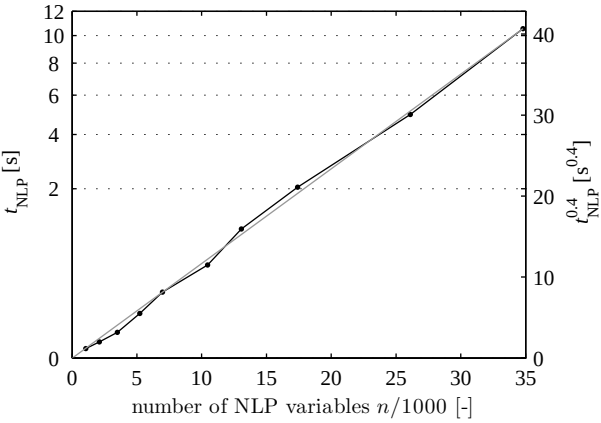

Figure 5.11: Runtime of SNOPT for the first three NLP iterations, without Jacobian evaluations, as a function of the number of NLP variables. The first three iterations are critical since the initial decomposition as well as most updates (or refactorisations) are performed during the initial phase of the solution. Test case 3 is used, and the step size of a uniform discretisation is changed such as to yield differently sized problems. This procedure allows the structure of the problem to be solved to remain the same, i.e. the number of active constraints and the degrees of freedom grow proportionally to the number of NLP variables.

warm-start capability is crucial, for which SNOPT is found to be the most suitable solver. Finally, the results are discussed and compared to the performance of the large-scale algorithm WORHP applied to the original problem.

Time decoupling As mentioned in Sec. 3.7.5, two specific characteristics of the OCP for diesel engines are exploited. First, the time horizon of the OCP is substantially larger than the characteristic time constant of the underlying dynamic system. In fact, the horizon of the OCP is in the range of 100 s, whereas the slowest relevant system dynamics,

which are imposed by the turbocharger inertia, exhibit a characteristic time of approximately 2 s.

Second, the state of the system is defined largely by the engine operating point. The fuel mass injected per cylinder and per combustion cycle is almost proportional to the load torque, which is prescribed. The fuel mass injected also defines the enthalpy introduced into the cylinders. From this energy, a portion of approximately 30% remains in the exhaust gas. The frequency of the combustion events is defined by the engine speed, which is a time-variable parameter in the OCP. Thus, the enthalpy flow through the turbine, which drives the turbocharger, is largely defined by the operating point prescribed by the driving cycle.

These two properties add up to an inherent decoupling of the state trajectories along the time axis. Thus, any demand for a certain end state only influences the last couple of seconds of the interval considered. The left column of Fig. 5.12 illustrates this finding. When just the interval up to 927 s is considered, the optimisation starts to open the VGT at around 925.5 s. Since the exhaust-manifold pressure drops faster than the boost pressure due to the slow turbocharger dynamics, a fuel saving results from the positive pressure drop over the cylinders.

From an optimal-control point of view, the co-state of a state variable without end constraint has to be zero at the end of the time interval. Zero penalty means that its influence on the subsequent time interval is neglected, i.e. no terminal cost is imposed.

Results First, the algorithms for solving the subdivided problem are analysed using test case 3. First-order collocation on a uniform grid with a step size of 0.1s is used, resulting in $N = 581$ steps and $n = 5229$ NLP variables for the original problem formulation. For the two-stage formulation, the time horizon is divided into 5 subintervals of 11.2 s

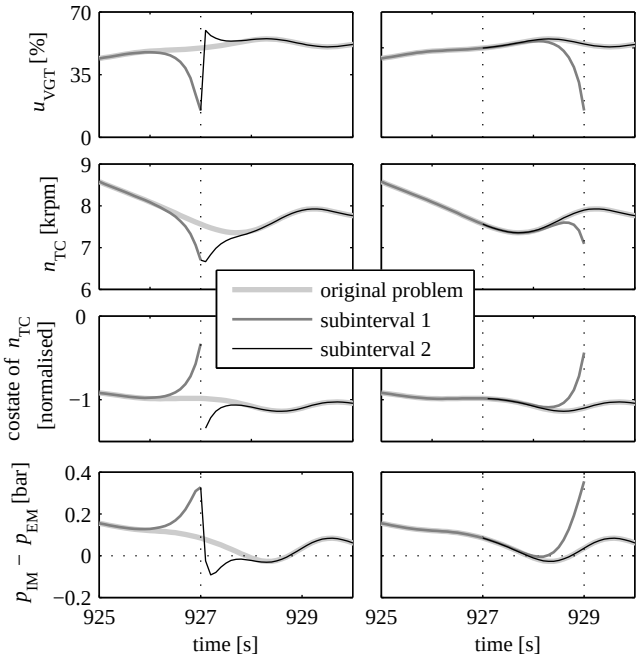

Figure 5.12: Region around an exemplary subinterval boundary; sequential solution without (left) and with overlap of 2 s (right). The solution of the original problem is displayed for reference. The partially dual formulation using the multipliers from the original solution is used for the subproblems to illustrate the equivalence after convergence of the high-level iteration.

length, and an overlap of 2 s is used. Each subproblem thus has $N = 133$ discretisation points, resulting in $n = 1197$ NLP variables.

Three high-level algorithms are tested: the one actually calculating the sensitivities by perturbing the emission limits (A), the approximation with three values for the tuning parameter $c = \{0.1, 0.2, 0.3\}$ (B.1-B.3), and the acceleration of the approximation with $n_{\text{acc}} = 3$ and $k_{\text{dec}} = 2$ (C). The sensitivity calculation within algorithm A uses a relative perturbation of the inequality limits of 1%. All algorithms are terminated whenever the following two criteria are met: a maximum relative violation of the emission limits of 0.05%, and a standard deviation of the subinterval-individual multipliers of less than 2%, relative to their mean value.

Figure 5.13 shows the relative fuel saving compared to a pre-series engine calibration, the relative violation of the emission limits and the standard deviation of the multipliers of the two emission species. Algorithm A converges in 5 iterations, whereas algorithms B.2 and B.3 converge after 9 or 7 iterations, respectively. The approximate algorithm with $c = 0.1$ (B.1) exhibits very slow convergence. The acceleration demonstrates its effectiveness by converging in one more iteration when using iterations 2 to 4 from B.1 to approximate the first-order sensitivities.

In order to validate the equivalence of the two-stage formulation and the original problem after convergence, the resulting trajectories of the control inputs and the state variables are compared. For all solutions obtained by the two-stage algorithms, they differ from the solution of the original problem by less than 0.5% of their allowable range.

Figure 5.14 presents the runtimes of the algorithms. The runtime per iteration is significantly larger for algorithm A than that of variants B and C. This increased computational time is due to the additional two

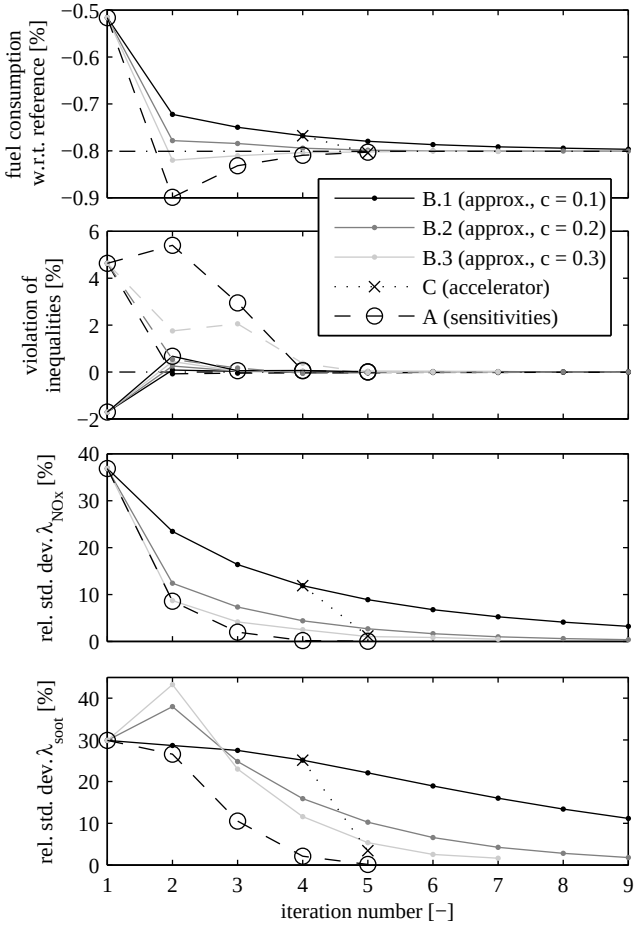

Figure 5.13: Results of the algorithms, test case 3: objective (fuel consumption related to the reference), violation of the constraints on the emissions (solid: NO_x, dashed: soot) and relative standard deviation of the emission multipliers. The dash-dotted line in the top plot is the objective value achieved by the original solution.

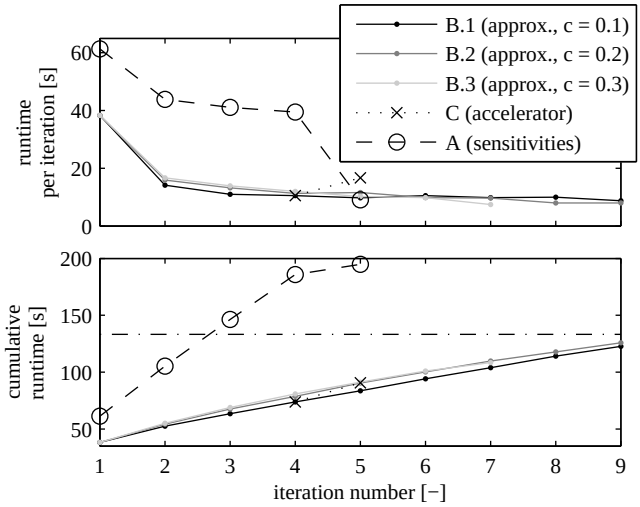

Figure 5.14: Results of the algorithms, test case 3: runtime per iteration and cumulative runtime. The dash-dotted line in the bottom plot is the runtime of the original problem.

NLP evaluations required to calculate the sensitivities. Except for the first iteration, the time spent for the solution of the two perturbed problems exceeds the time required by the original optimisation. During later iterations, the actual change in the limits becomes similarly small as the perturbation used in the sensitivity calculation. The runtimes of the approximate algorithms B.1–B.3 are slightly shorter than the one of the original problem. The acceleration C saves a significant amount of time by reducing the number of iterations to achieve convergence.

For the realistically dimensioned test case 7, the time subdivision of the OCP is combined with an iterative mesh refinement. Two refinements are performed, both with a goal of a relative tolerance of 0.01. The first one uses a safety margin of $k = 0.5$, i.e. new points are added cautiously, whereas the second refinement is applied with $k = 1$. When

no mesh refinement is applied, a fixed step size of 0.075 s is chosen. This discretisation provides a similar accuracy of the solution, and yields a total of $N = 3934$ steps and $n = 35{,}406$ NLP variables.

The two high-level algorithms A and B.2 are tested. The choice of $c = 0.2$ for the approximate iteration is found to represent a reasonable tradeoff between stability and convergence speed for large and less homogeneous problems such as test case 7. For the iterative algorithms, a mesh refinement is performed whenever the relative standard deviation of the multipliers is below 10% and the violation of the inequality limits is less than 2%. For algorithm A, no sensitivities are calculated for this iteration but only the mesh is refined. For algorithm B.2, an approximate update as well as a mesh refinement are performed for the next iteration.

Table 5.4 lists the number of high-level iterations, the number of grid points and NLP variables, and the runtimes. As for test case 3, the additional cost to evaluate the sensitivities outweighs the advantage of a smaller number of iterations when algorithm A is compared to algorithm B. In fact, the time required for the solution of perturbed problems accounts for 46% of the total runtime of the subdivided OCP. The reduction in computation time is larger when time subdivision is applied, compared to when mesh refinement is used alone. When time subdivision and mesh refinement are combined, their individual potential for reducing the computation time cumulates.

For test case 7, algorithm C is not able to cut the number of iterations required to attain convergence of the high-level iteration. One shortcoming of the approach proposed is the fact that the model for the relation between the cumulative inequality limits and their Lagrange multipliers does not adapt its complexity in accordance with the availability of more data. Thus, it is not able to reproduce local nonlinearities of this relation. However, for subintervals comprising long drag phases,

Table 5.4: Results of the algorithms, test case 7: absolute and relative runtimes t_{tot} (values without parallelisation in brackets), number of discretisation points N, number of high-level iterations HLi, and parallel speedup factor PSUF. In accordance with Fig. 3.4, full parallelisation is used for the original OCP, and model-only parallelisation is applied to the subproblems of the two-stage algorithms.

algorithm	no refinement	with ref.
absolute runtime t_{tot} [s] (no para.)		
original	13,217 (13,422)	1,906 (2,466)
A	1,419 (2,321)	897 (2,231)
B	1,305 (2,134)	519 (1,188)
relative runtime t_{tot} [%] (no parallelisation)		
original	98.47 (100.00)	14.20 (18.37)
A	10.58 (17.30)	6.69 (15.13)
B	9.72 (15.90)	3.86 (8.85)

N, (PSUF) [HLi]

algorithm	no refinement	with refinement
original	3,934 (1.02) [1]	2,029 (1.29) [3]
A	3,934 (1.64) [5]	2,027[a] (2.26) [10]
B	3,934 (1.64) [9]	2,026[a] (2.29) [12]

[a] For the subproblems, N is between 77 and 181.

the absolute mass of cumulative emissions is small and therefore, the multipliers are very sensitive to changes of these limits.

Discussion The approach proposed reduces the time to solve an OCP over a long time horizon by exploiting two properties in a combined way. On the one hand, the superlinear increase of the computational time as a function of the problem size, as shown in Fig. 5.11, is mitigated by subdividing the original problem into smaller subproblems. On the

other hand, the construction of the Jacobian starts to dominate the cost for the solution of small NLPs, see Fig. 3.7 (top). Thus, by using shared-memory parallelisation during the calculation of the Jacobian, a larger speedup factor of the full NLP solution can be achieved, see Fig. 3.7 (bottom).

For the OCP of a diesel engine over a prescribed driving cycle, the parameter space of the high-level optimisation can be reduced. The inherent time separation of the state trajectories allows for a sequential solution of the subdivided OCP with only a short overlap phase, as demonstrated in Fig. 5.12. Thus, the system state at each subinterval boundary is eliminated from the high-level optimisation while the small size of the subproblems is preserved.

The remaining task of the high-level optimisation consists of the allocation of the net allowable cumulative emissions to the subintervals such as to achieve equivalence with the original problem formulation. The partially dual formulation of the subdivided OCP reveals the conditions for equivalence. Furthermore, the direction as well as the step size of the high-level iterations are well defined by the sensitivities of the dual variables to the emission limits and the global inequality limits. Thus, a fast convergence of the high-level problem is ensured. In order to eliminate the costly calculation of sensitivities, an approximation for the local sensitivities is proposed. The slower convergence, which leads to a larger number of high-level iterations, is outweighed by the fact that the sensitivities do not have to be explicitly calculated.

The two-stage approach is shown to perform well for a problem with a realistic size and structure. A speedup factor of 14 is achieved in the numerical experiments. Aside from the implementation of the high-level updates and a straightforward parallelisation of the Jacobian calculation, no custom code is required. The runtime of the entire solution procedure is linear in the time horizon considered. Appending

additional intervals to the subdivided problem does not necessitate more high-level iterations since the problem structure does not change substantially along the time axis.

An iterative refinement of the discretisation grid has a similar effect on the problem size as the time-domain decomposition. On the one hand, the average size of the problem is smaller since a coarse mesh is used initially. On the other hand, the size of the final problem is smaller since smaller step sizes are only applied where necessary. This effect is evident in Table 5.4.

Since the solution of the two-stage problem as well as the mesh refinement are iterative processes, they can be combined easily. The individual speedup factors of the two techniques cumulate such as to yield an overall speedup factor of up to 25 in the realistic test scenario.

Many extensions to the approach presented can be thought of. For algorithm A, the subproblems could be initialised by utilising the sensitivities of the discretised control inputs and state variables which are obtained anyway during the sensitivity calculation. However, the active set, i.e. the time intervals during which some constraint is active, changes. Thus, the initialisation by the sensitivities does not always yield a closer guess for the solution. First tests show promising results whenever the active set does not change substantially between two subsequent high-level iterations.

Parametrisation of the solution, and thus of the active set, could make use of the information obtained during previous high-level iterations. In fact, the accelerator proposed for the approximate iteration is a first step in that direction. The first-order model used here, however, is limited to a local approximation of the relation between the NLP variables of the subproblem and the high-level variables. An increasingly complex model could be constructed, necessitating additional solutions of the subproblem only if the model cannot be trusted in

that region as yet. Thus, the number of NLP evaluations necessary for convergence of the high-level optimisation could be reduced. Furthermore, a parametrised solution as an additional output of the iterative procedure may be helpful for some applications, e.g. for an online application of the optimal control trajectories on a system with disturbances or changing parameters.

Finally, a more custom-tailored NLP solver could obtain the sensitivities more efficiently. For example, the matrix factorisation of the solution to the unperturbed NLP could be used such that only few updates of the factorisation would be necessary when the perturbed NLP is solved.

As stated in Sec. 5.1.1, WORHP achieves a runtime that is comparable to the one of the time-subdivision approach. Also for the large-scale approach, the key is to combine an iterative mesh refinement with the warm-start capability of an SQP solver. Since exact-Newton methods are found to be most efficient in solving large-scale problems, the reduction of the runtime by a faster execution of the model functions is larger than for the time-subdivision approach, which can use a QN solver that is efficient for the solution of the relatively small subproblems.

5.1.6 Conclusions

On the simple test case 1, all discretisations and all methods to solve the resulting NLPs seem to work reasonably well. However, especially the QN methods converge faster for a local discretisation than for the pseudospectral approach. The trust-region method implemented in KNITRO always gets stuck at a small trust radius and thus converges only slowly towards the optimum. The line-search globalisation thus seems to be preferable for the problem at hand.

When more meaningful driving profiles such as test cycles 3 and 4 are considered, the QN methods become less efficient also for local

discretisation schemes. Surprisingly, SNOPT still manages to solve most problem instances within 200 outer iterations. As the complexity and the problem size increases, the SQP method implemented in WORHP becomes more reliable and consistent than the IP method of IPOPT.

The pseudospectral approach yields a less sparse NLP as indicated in Fig. 3.3. Although the Hessian of the Lagrangian is still block diagonal, the Jacobian of the constraints does not retain a near-diagonal shape. The decomposition performed by SNOPT as well as the direct linear solvers of the full-matrix approaches become disproportionally slow when the collocation order is increased but the problem size is not changed.

Combined with the effectiveness of a step-size refinement, local discretisation schemes seem to be preferable for the problem at hand. A local discretisation enables a finer resolution of the problem only where necessary, and an SQP solver is able to exploit the good initialisation of the refined problem. Conversely, the pseudospectral method can only increase the order of the collocation polynomial and thus always refines the approximation of the problem over the full time horizon.

The regularisation implemented in the framework is found to be an effective means to enforce smooth control trajectories. Since the numerical problems do not originate from an insufficient accuracy of the ODE solution, a finer approximation of the state than of the control inputs, i.e. multiple shooting, is not preferable for the problem at hand. However, the phenomenon of the spurious oscillations over singular arcs should be analysed in depth to derive a more targeted strategy to resolve this problem.

Summarising these findings, a full-matrix approach for the solution of the KKT system utilising a direct linear solver should be combined

with an exact-Newton SQP method and a line search globalisation.[2] WORHP implements such a method, and in fact this solver is found to perform well on all test cases, as well as to adapt to large problems best. A relatively low-order collocation scheme and an iterative step-size refinement combine well with this type of NLP solver. Only for small problems arising in receding horizon control, for instance, a decomposition approach and a QN method such as that implemented in SNOPT prove to be more efficient.

A full-matrix, exact-Newton SQP method such as WORHP is able to efficiently solve large-scale problems. Therefore, future work should focus on speeding up the derivative calculation or the evaluation of the model functions in general. At the same time, the separability of the transcribed problem should be exploited to construct a more accurate approximation of the exact Hessian by a partitioned quasi-Newton update. The effectiveness of such an approach is indicated by the preliminary results presented in Sec. 5.1.3. A convergence behaviour close to that of an exact-Newton method is achieved, but only first-order derivatives are required. An integrated implementation in an NLP solver could fully exploit the potential of this approach. In contrast to these further improvements possible for large-scale algorithms, the potential of the time-subdivision approach is almost exhausted, except for a more efficient calculation of the NLP sensitivities.

[2] The effect of different merit functions or a filter was analysed, too. No unique trend could be observed that favours one specific approach. The problem at hand thus seems not to be susceptible to the Maratos effect [143, Sec. 15.5].

5.2 Engineering aspects

This section first presents the results from a single iteration of the iterative approach to the dynamic optimisation of diesel engines. Subsequently, three ways of utilising the solutions of the optimal control problems are proposed. First, it is shown that optimal control can be used to assess the effects of performing a fully or partially static calibration of the engine only. Second, the optimal solution may be used to identify all maps and even the control parameters of an entire feedback-control structure. Third, control strategies for transient operation can be derived from parametric studies. The latter is illustrated by a case study in which the NO_x emissions of an engine with EGR have to be reduced over a load increase.

5.2.1 Transient model refinement

The initial iteration of the iterative approach to dynamic optimisation proposed in Sec. 4.5 is analysed in this section. The transient model refinement is illustrated, and the properties of the time-variable combustion model are elaborated. Throughout this section, engine D is considered.

A short test cycle is used to evaluate the various methods. The engine is operated at a constant speed of 2500 rpm, and the desired

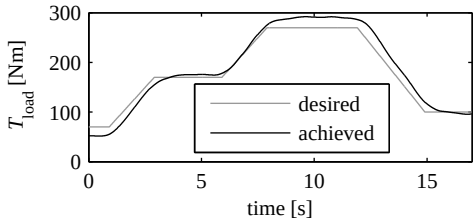

Figure 5.15: The desired load-torque profile and the trajectory achieved by the testbench controller.

load-torque profile is shown in Fig. 5.15. As variations, an additive offset of ±5% PWM is applied to the VGT and EGR positions, and ±2° to the SOI. A multiplicative factor of 1.05 defines the variation of the fuel mass. These variations are referred to as the *small* variations in the remainder of this section. To assess the accuracy of interpolation and extrapolation, *large* variations with offsets of ±8% for the VGT, ±10% for the EGR, ±4° for the SOI and a factor of 1.1 for the fuel mass are performed. Since the VGT and the EGR both dynamically affect the burnt-gas fraction and the pressure in the intake manifold, the four *cross variations* of these two actuators are additionally recorded.

For the identification of the combustion model, it would be desirable to have a constant offset in $\boldsymbol{v}$. However, the EGR-VGT controller cannot perfectly follow reference trajectories for the burnt-gas fraction and the boost pressure. After initial tests, it has been found that applying constant offsets directly to the two dynamic controls is the best choice.

Transient air-path model refinement The turbocharger speed and the pressure in the exhaust manifold represent the relevant dynamics in a turbocharged engine system with EGR. The boost pressure closely follows the turbocharger speed and the EGR mass-flow is defined, aside from the position of the EGR valve, by the pressure ratio between exhaust and intake manifolds. Therefore, it is sufficient to correct these two dynamics.

Since for the combustion model only the *relative* accuracy is of interest, it is not necessary to require $\boldsymbol{v}(t)$ to be accurate in terms of absolute values. Rather, the state variables that are critical for a safe engine operation should be matched to the measured trajectories to enable an accurate limitation of these quantities in the OCP. Since no limits on any state variables are included in the OCP for engine D as

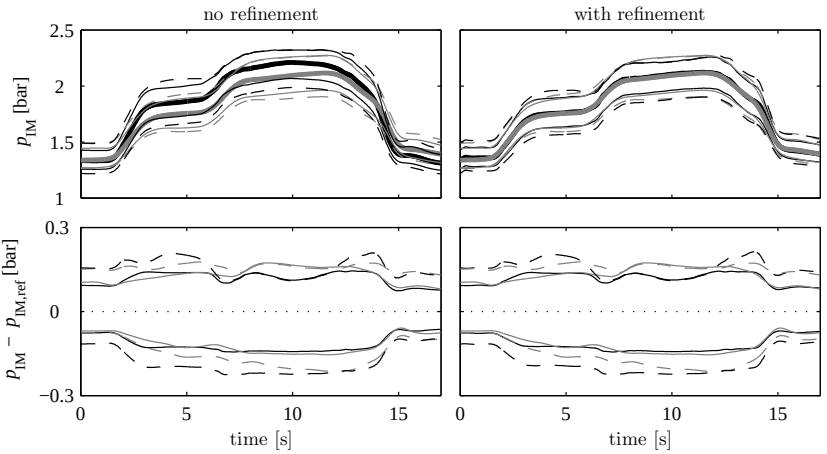

Figure 5.16: Effects of the dynamic model refinement, VGT variations. Measurement data (grey) versus model outputs (black). Line styles: reference (bold), small variations (solid), large variations (dashed).

of yet, the pressures in the exhaust and intake manifolds are chosen to illustrate the methodology. The choice of the dynamics and the state variables to be corrected is represented by

$$\boldsymbol{K}_{\mathrm{f}}^{T} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{K}_{\mathrm{g}}^{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \quad (5.1)$$

The errors in the two pressures are equally weighted, i.e. $\boldsymbol{b}^{T} = (1\ 1)$.

The top plot in Fig. 5.16 shows the measured trajectories of the intake pressure along with the model output before and after the refinement. The refinement is performed using only the small variations. The resulting good match for the large variations indicates that the model errors in fact are systematic. For example, the too fast speedup of the turbocharger predicted by the model might be caused by the omission of the thermal models. The heat losses to the manifold walls

are neglected and consequently, the enthalpy available to the turbine is overestimated.

The bottom plot shows the difference of the variations to the reference trajectory. Obviously, the refinement has no influence on the predicted difference. Therefore, the refinement of the air-path model is not critical for the combustion model but rather a tool that allows for an accurate limitation of any state variable, e.g. maximum turbocharger speed, exhaust-manifold pressure and temperature, etc.

In the context of the generalised Kalman filter, the regularisation described in Sec. 3.8 may be used to enforce smooth trajectories of the corrective variables $\tilde{\boldsymbol{x}}$. The demand for smooth trajectories is justified by assuming the model errors to be of systematic nature and thus not to exhibit a stochastic or arbitrarily fast changing behaviour. For the results presented here, a value of $c_{\mathrm{reg}} = 50$ is used.

Static combustion model The results presented here are all derived using the time averaging introduced in Sec. 4.5.2. Interpolation and extrapolation is not influenced by the time averaging. Table 5.5 shows the errors of the combustion model when identified using three different data sets, namely the small variations, the small and the cross variations, and the large variations. The figures indicate that the model identified using the small variations is able to accurately predict the cross variations, except for the soot emissions. Furthermore, interpolation is significantly more reliable than extrapolation. This fact encourages the use of rather large variations and small factors k_{u} and k_{x} instead of the reliance on small variations and extrapolation.

Optimal control For the OCP, the regularisation was set to $c_{\mathrm{reg}} = 100$. This choice successfully suppresses oscillating solutions while not affecting the parts that exhibit smooth trajectories anyways. Especially

Table 5.5: Static combustion model: average magnitude of the relative error in % for the instantaneous NO_x and soot emissions, and the torque T_{load}.

ident. data:	*small variations*			*small & cross vars.*			*large variations*		
	NO_x	soot	T_{load}	NO_x	soot	T_{load}	NO_x	soot	T_{load}
small vars.:	0.14	0.89	0.09	0.70	4.45	0.25	1.14	7.70	0.75
cross vars.:	2.85	15.84	0.82	1.60	12.41	0.27	2.60	18.92	0.73
large vars.:	3.31	20.56	2.36	3.46	21.17	2.11	0.19	1.10	0.09

for the SOI, which has no influence on the air-path dynamics, fast oscillations result when no regularisation is applied. The time averaging during the identification of the combustion model has a similar effect as the regularisation. It is left to be analysed in more detail which approach provides the most plausible solutions and the fastest convergence of the iterative procedure.

Figure 5.17 shows the optimal control trajectories obtained when using the model identified by the small variations only, or by the small and the cross variations. In both cases, no extrapolation is performed, i.e. $k_u = 1$ and $k_x = 1$. The limits for the cumulative emissions, $\hat{\boldsymbol{m}}_{em}$, are chosen such that the brake-specific emissions of the reference are maintained. The experimental validation of the two optimal solutions is presented in Sec. 5.3.2 below.

5.2.2 From static do dynamic optimisation

First, the concept of partially static optimal control is introduced. Subsequently, this methodology is applied to engine A to illustrate the effects when certain dynamics of the air path are neglected. Test cycle 7 is used throughout this section.

Partially static optimal control Treating a state variable as an algebraic variable is equivalent to setting its inertia, e.g. the volume of a

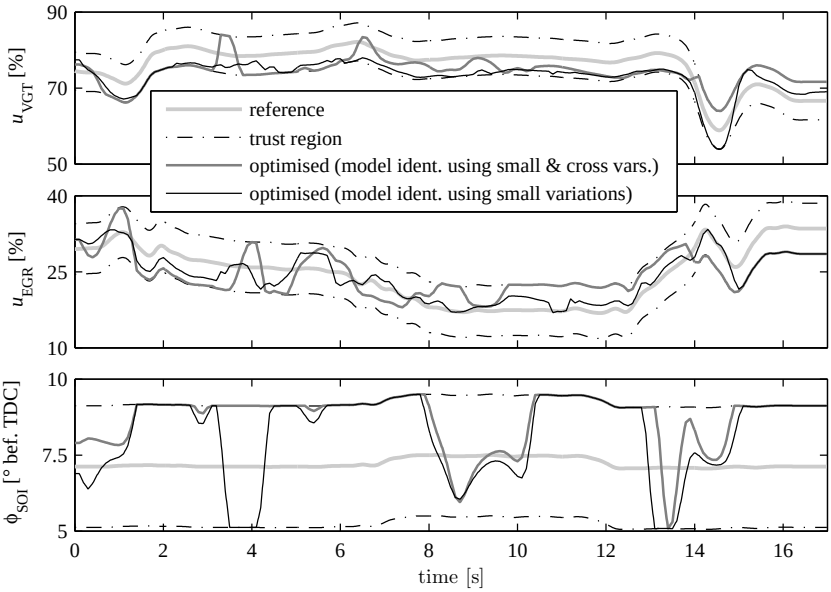

Figure 5.17: Control trajectories resulting from the solution of the OCP constrained by the model-validity region.

receiver or the rotational inertia of the turbocharger, to zero. Thus, the system of ODEs describing the model, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\pi})$, is transformed into a system of differential-algebraic equations

$$\begin{pmatrix} \dot{\boldsymbol{x}}_\mathrm{d} \\ 0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_\mathrm{d}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\pi}) \\ \boldsymbol{f}_\mathrm{s}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\pi}) \end{pmatrix}. \tag{5.2}$$

The state variables are split into a dynamic set $\boldsymbol{x}_\mathrm{d}$ and a static set $\boldsymbol{x}_\mathrm{s}$ with $\boldsymbol{x} = (\boldsymbol{x}_\mathrm{d}^T, \boldsymbol{x}_\mathrm{s}^T)^T$. This type of model is difficult to simulate since implicit nonlinear equations need to be solved along with the integration of the ODEs.

In contrast, in the context of direct transcription, adopting this structural change is straightforward. The dynamic constraints are divided into two sets. For example, for first-order Radau collocation, the two sets of constraints on each collocation interval read

$$\boldsymbol{x}_{\mathrm{d},k+1} - \boldsymbol{x}_{\mathrm{d},k} - h_k \cdot \boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x}_{k+1}, \boldsymbol{u}_{k+1}, \boldsymbol{\pi}_{k+1}) = 0, \tag{5.3a}$$

$$\boldsymbol{f}_{\mathrm{s}}(\boldsymbol{x}_{k+1}, \boldsymbol{u}_{k+1}, \boldsymbol{\pi}_{k+1}) = 0. \tag{5.3b}$$

This slight change of the problem hardly affects the iterations of the NLP solver. In fact, the problem becomes more time separated since the discretised static state variables are no longer coupled to the neighboring discretisation points.

Four steps from a dynamic to a static optimisation are considered.

1. *Dynamic optimisation*, i.e. optimal control. The original OCP for diesel engines is solved.

2. *Static pressures.* The two pressure dynamics, which are the fastest in the system, are neglected. This simplification is equivalent to setting the volumes of the intake and exhaust manifolds to zero.

3. *Static air path.* In addition to step 2, the turbocharger inertia is set to zero. Therefore, the turbocharger lag is neglected.

4. *Static optimisation.* In addition to step 3, the thermal dynamics of the engine are assumed to always be in steady state. This step emulates the common procedure of operating the engine at a certain operating point for several minutes before recording the data for the calibration.

Step 4 is equivalent to a representation of all state variables and control signals as maps over engine speed and load. However, the number of optimisation variables is larger in the time-discretised problem.

In fact, during a long driving cycle, the same operating point occurs several times, introducing redundant variables. Nevertheless, for two reasons the optimal-control framework presented here is used also for this case. First, the same discretisation grid is used as for the dynamic cases, which allows for a consistent comparison of the resulting maps. Second, the same integration scheme and thus the same weighting of the operating points is applied. If an operating-point discretisation is used instead of a time discretisation, a different problem would be solved, which would further impair the comparability to the dynamic case.

Results from the optimisation In this paragraph, the output of the four optimisation cases is studied and the fuel consumption predicted by each case is stated. In the next paragraph, the optimal solutions are used in a forward simulation of the dynamic model. The simulation results are analysed and compared to the prediction of the corresponding optimisation presented here.

The fuel consumption and the pollutant emissions predicted by the four optimisation approaches are listed in Table 5.6. All the partially static cases achieve a lower fuel consumption since they are less severely restricted due to the reduced system dynamics. The emission limits are perfectly met by all the optimal solutions.

In order to assess whether a quantity is “quasi-stationary”, the following procedure is applied. A surface spanned over the engine speed and the load torque is fitted to the data resulting from the optimisation. If the data can be represented perfectly by this map, this quantity is related algebraically to the operating point and can be considered a quasi-stationary variable.

Figure 5.18 shows an illustrative example. The VGT position resulting from the four optimisation approaches is plotted along with the

Table 5.6: Fuel consumption and pollutant emissions predicted by the optimisation and obtained by simulation using the optimal solution. The fuel consumption is related to case 1 and the pollutant emissions are related to their respective limits.

case number	*1*	*2*	*3*	*4*
Optimisation				
fuel w.r.t. (*) (%)	(*)	99.58	99.88	99.89
NO_x w.r.t limit (%)	100.00	100.00	100.00	100.00
soot w.r.t limit (%)	100.00	100.00	100.00	100.00
Dynamic validation				
fuel w.r.t. (**) (%)	(**)	100.21	101.41	102.80
NO_x w.r.t limit (%)	99.93	96.56	94.73	84.71
soot w.r.t limit (%)	100.97	96.01	101.72	145.03

surfaces fitted to the respective data. For the fully stationary case 4), the data can be considered to be a function of the engine speed and load torque. For case 3), where only the air-path is considered to be static, a similar reasoning holds since only the slow thermal dynamics are included in the model. However, the VGT is slightly more closed for high loads due to the reduced enthalpy flow in the exhaust. For the dynamic case 1), the data does not lie on a surface. The VGT has to be actuated aggressively during transients in order to achieve a sufficiently fast buildup of the boost pressure. Therefore, the static relation to the operating point is canceled by the dynamics of the system. Case 2) lies in-between, since only the fast pressure dynamics are omitted. However, the air-path dynamics are mainly defined by the turbocharger inertia, which still is present.

The standard deviation of the optimisation data from a surface is listed in Table 5.7 for all optimisation approaches and for all control inputs and state variables. For a more intuitive reading, the deviation

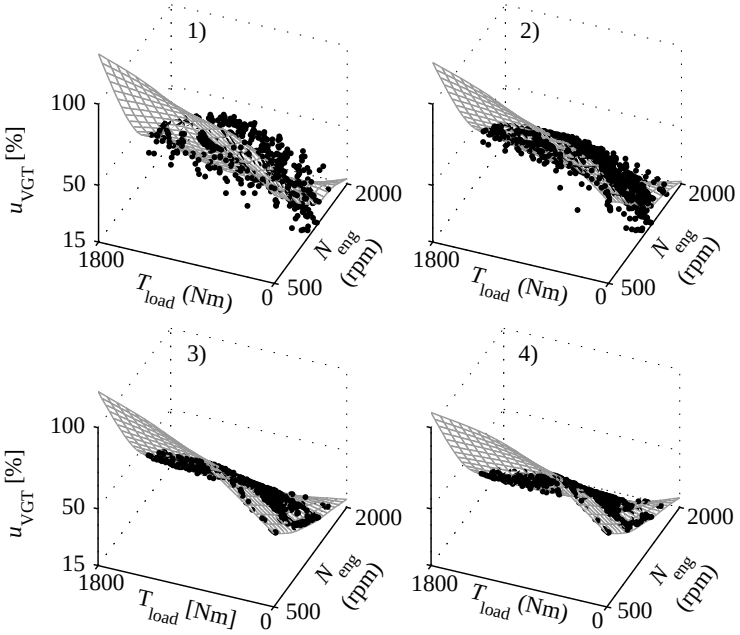

Figure 5.18: VGT control inputs resulting from the four optimisation approaches and the surfaces fitted to that data.

related to the range of the data is provided along with the absolute values.

Dynamic validation of the results To assess the effects of neglecting certain dynamics, the solution of the OCP is used in a forward simulation of the dynamic model. The same driving-cycle segment is considered as is used during the optimisation. From the simulation data, the cumulative fuel consumption and the pollutant emissions are evaluated. They are listed in the lower part of Table 5.6. The dynamic optimisation (case 1) accurately predicts the fuel consumption and the cumulated emissions.

Table 5.7: Standard deviation of the optimisation data from the fitted surface. The relative values are normalised by the range of the data from the dynamic optimisation (case 1).

case number	*1*	*2*	*3*	*4*
VGT position (% PWM)	8.67	4.95	0.70	0.53
→ *relative (% of range)*	*(12.17)*	*(6.95)*	*(0.99)*	*(0.74)*
start of injection (°)	0.29	0.22	0.18	0.12
→ *relative (% of range)*	*(2.40)*	*(1.82)*	*(1.47)*	*(1.01)*
rail pressure (bar)	19.12	14.93	12.63	12.90
→ *relative (% of range)*	*(1.76)*	*(1.38)*	*(1.17)*	*(1.19)*
intake-manif. pressure (kPa)	6.35	3.72	1.69	1.69
→ *relative (% of range)*	*(3.23)*	*(1.90)*	*(0.86)*	*(0.86)*
exhaust-manif. pressure (kPa)	11.15	5.33	1.68	1.40
→ *relative (% of range)*	*(5.71)*	*(2.73)*	*(0.86)*	*(0.72)*
turbocharger speed (krpm)	2.94	2.22	1.07	1.00
→ *relative (% of range)*	*(3.14)*	*(2.38)*	*(1.14)*	*(1.07)*
wall temperature, exhaust (K)	39.45	39.64	38.84	10.49
→ *relative (% of range)*	*(6.95)*	*(6.98)*	*(6.84)*	*(1.85)*
wall temperature, intake (K)	0.48	0.48	0.45	0.11
→ *relative (% of range)*	*(2.38)*	*(2.37)*	*(2.24)*	*(0.53)*

Brake-specific values, i.e. quantities in units of (g/kWh), are used for the fuel consumption and the emissions. Relating the cumulative masses of fuel and emissions to the integral of the net power produced by the engine allows for a fair comparison of the four cases. When the air path is treated as being static, the transient pressure difference over the cylinders and the resulting loss of usable torque are neglected. Therefore, a lower net torque than desired results during the dynamic simulation using the resulting optimal control trajectories. This effect is inherently accounted for when brake-specific quantities are used.

All optimal control solutions obtained with a partially static model generate a higher fuel consumption during the dynamic validation. Furthermore, the emission limits are not matched exactly. Two effects can be observed. First, the more static the optimisation, the higher the dynamically calculated soot emissions. Second, the NO_x emissions show the opposite trend, while the fuel consumption increases. These two effects are explained and illustrated next.

Figure 5.19 explains the increasing soot emissions. It shows the VGT position, the boost pressure and the instantaneous soot emissions over the first 24 seconds of the segment of the WHTC considered. The more static the optimisation, the longer the "delay" of the VGT actuation becomes. This hesitation results in too low a boost pressure and thus in an insufficient cylinder charge. The lack of oxygen leads to peaks in the soot emissions, as the shape of the soot surface in Fig. 2.44 clearly shows. During the interval of 24 seconds shown, the solution from the static optimisation (case 4) produces 4.13 times the soot emissions allowed.

There exists an inherent tradeoff between the fuel consumption and the NO_x emissions. A fast and hot combustion is efficient, but it also favours the thermal formation of NO_x. Figure 5.20 reveals that the static optimisation overestimates the thermal state of the engine. In order to compensate for the higher NO_x emissions due to the higher temperature of the cylinder charge, the optimisation retards the start of injection. The combustion becomes slower and cooler, but also less efficient. During the dynamic simulation, the engine never heats up to fully stationary conditions since this process would take several minutes. Therefore, the charge temperature is lower and the NO_x-fuel tradeoff is strongly biased towards low NO_x emissions. During the interval of 24 seconds shown in the plot, the solution from the static optimisation leads to a fuel consumption that is 3.92% higher than the one achieved by the dynamic optimisation.

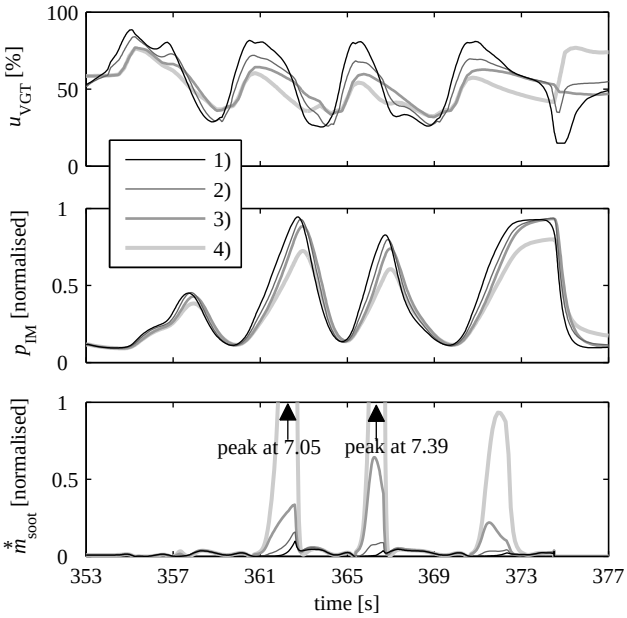

Figure 5.19: Results of the dynamic validation, first 24 seconds of test cycle 7. A retarded action of the VGT during transients results in a lack of boost pressure and thus an insufficient cylinder charge. The low air-to-fuel ratio leads to peaks in the soot emissions. For reasons of confidentiality, normalised values are shown.

Discussion The results presented allow several conclusions to be drawn. The results listed in Table 5.7 imply that it is optimal to represent the control signals defining the combustion, namely the start of injection and the common-rail pressure, by maps over the engine speed and load. The dynamic optimisation, which at each time instant is provided the full freedom of arbitrarily choosing the control inputs, generates almost perfect lookup maps, as indicated by the low variance w.r.t. the surface fitted to the data.

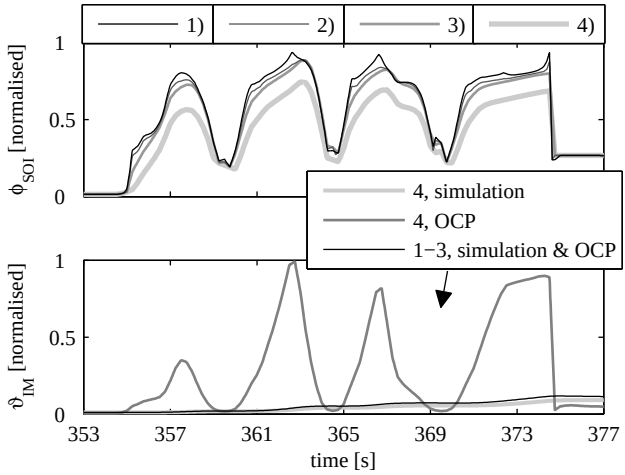

Figure 5.20: Statically calculated intake-manifold wall temperatures (during the OCP for case 4) and the trajectories from the dynamic validation (bottom). In case 4, the optimisation assumes too high temperatures and compensates the increased NO_x emissions by retarding the start of injection (top). For reasons of confidentiality, normalised values are shown.

Similarly, the optimal intake pressure can be closely represented by a lookup map over the operating range of the engine. Therefore, always achieving the value prescribed by this map would be close to optimal. Establishing this matching, however, is exactly the task of the feedback control of the intake pressure by actuating the VGT. The even lower variance of the turbocharger speed indicates that using this quantity as the feedback signal would be preferable.

For this control loop, the VGT map obtained by case 3 (static air path, see Fig. 5.18), can be used as optimal static feedforward control signal u_{FF}. The optimal dynamic control signal u as well as the optimal trajectories of all potential feedback signals are known from the

dynamic optimisation (case 1). Thus, the optimal feedback control signal $u_{\mathrm{FB}} = u - u_{\mathrm{FF}}$ is known as well. By utilising this knowledge, the development of the feedback controller can be isolated from the stationary engine calibration, and the solution of case 1) serves as a means to benchmark different control designs.

An optimal choice of the feedback signals could be obtained by the following procedure. Apply the dynamic optimal VGT position, but use the static optimal maps for all other inputs. The resulting state trajectory that most closely matches its dynamic optimum is suited best as a feedback variable. Especially for MIMO systems, the best coupling of the feedback signals to the actuators, and the degree of dynamic cross influences could be assessed by this analysis.

The mismatch between the prediction of all partially static optimisation approaches and the dynamic simulation listed in Table 5.6 emphasises the necessity of considering the engine dynamics during the engine calibration. In particular, Fig. 5.20 implies that only a steady state of the air path but not of the thermal dynamics should be reached when measurements are performed for the identification of empirical models or for the application of statistic calibration tools.

5.2.3 Model-based dynamic engine calibration

Throughout this section, engine A is considered. From the solution of the OCP over a sufficiently long time horizon such as test case 7, implications for the control structure can be derived as proposed in the previous section. The optimal trajectories of the control inputs defining the combustion, i.e. the SOI and the rail pressure, can be represented accurately by static maps over the engine operating range. The same finding applies to the boost pressure. Although these quantities might be chosen freely over time by the optimisation, values result that can be scheduled over the engine speed and the injected fuel mass. If the

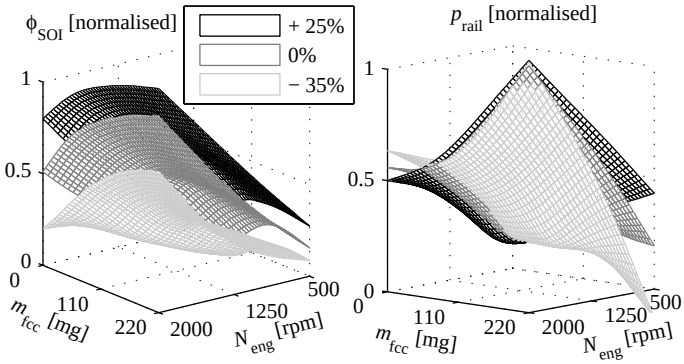

Figure 5.21: Maps obtained from the optimal control trajectories, different NO_x levels relative to the one resulting from the current ECU calibration. Normalised signals are shown for reasons of confidentiality.

air path is considered to be stationary, a static feedforward map for the VGT is obtained from the corresponding OCP.

Two applications of these findings are presented here. First, the maps for the combustion control can be derived for different emission levels. The maps for the SOI and the rail pressure for three different NO_x levels are shown in Fig. 5.21. These maps can be parametrised by the requested emission level. On the ECU, the maps could be shifted adaptively, depending on the current performance of the ATS or according to the current driving situation.

The second application is to implement a feedback controller for the boost pressure based on the maps for the boost pressure and the static optimal VGT position. The former serves as reference to be tracked by the controller, and the latter is used as feedforward control signal. Figure 5.22 shows the structure of the control system. During drag phases, a pure feedforward controller is applied. The map for the VGT position during motored operation is spanned by the engine speed and

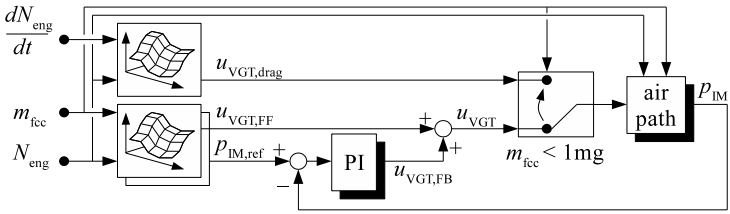

Figure 5.22: Structure of the boost-pressure controller.

its derivative, and is identified using the optimal data from the last 2 seconds of each drag phase [22]. Along with this controller defining the VGT position, the maps presented in Fig. 5.21 are used for the SOI and the rail pressure.

The PI controller for the VGT is implemented as

$$u_{\mathrm{VGT,FB}} = k_{\mathrm{p}} \cdot e_{p_{\mathrm{IM}}}(t) + \frac{k_{\mathrm{i}}}{10} \cdot \int_0^t e_{p_{\mathrm{IM}}}(\tau)\, d\tau, \tag{5.4}$$

with $e_{p_{\mathrm{IM}}} = 10^{-4} \cdot (p_{\mathrm{IM}} - p_{\mathrm{IM,ref}})$. The scaling factors are used to provide a similar magnitude of the two controller parameters. A classical anti reset-windup scheme [117] is applied to handle actuator saturation and the purely feedforward operation during drag phases.

The two PI parameters are varied on a reasonable grid, and a forward simulation of the model is performed for all combinations. Figure 5.23 displays the results. The NO_x emissions are rather insensitive w.r.t. the choice of the PI parameters. Similarly, the soot emissions increase rapidly only if a slow feedback controller is used. In this case, the boost pressure is too low during load increases, resulting in low AFRs. An aggressive controller yields the lowest fuel consumption. However, the control signal overshoots and oscillates for this parameter set. Therefore, the parameter set which yields the closest representation of the optimal control-input trajectory is chosen.

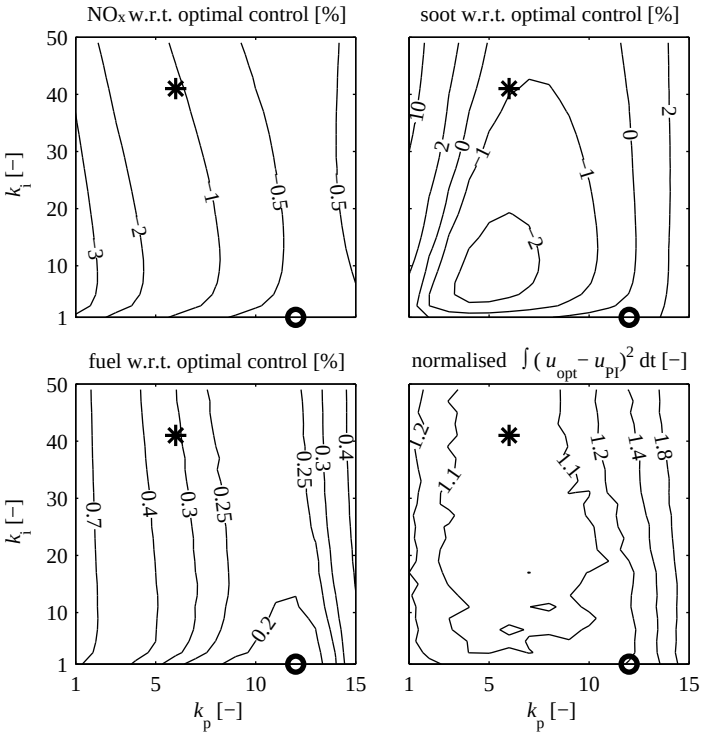

Figure 5.23: Selection of the parameter values for the PI controller. The circle indicates the fuel-optimal pair, while the asterisk denotes the values that result in the closest possible approximation of the optimal control-input trajectory.

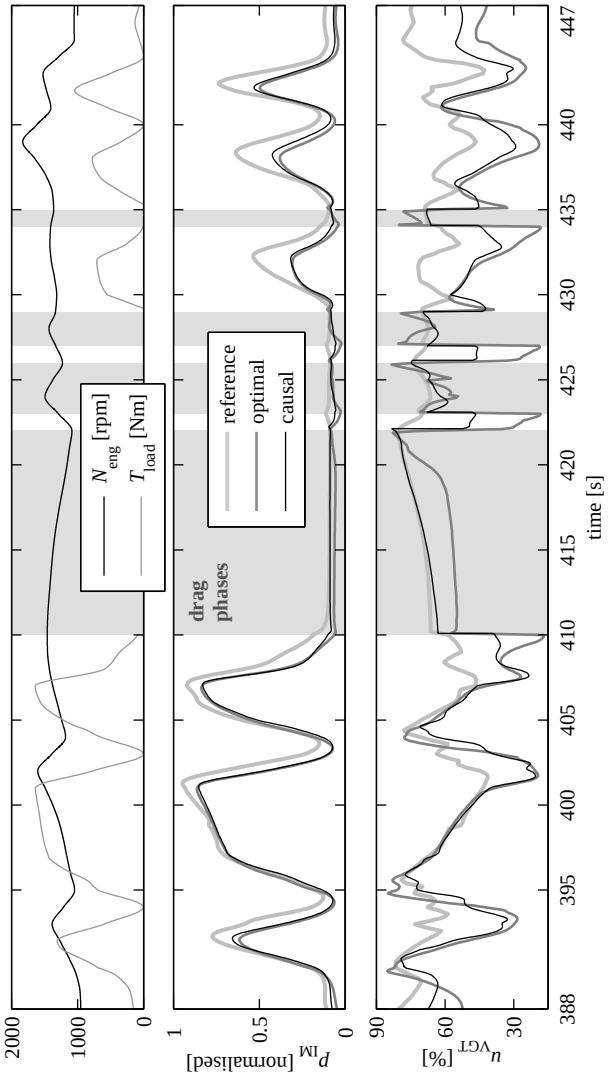

Figure 5.24: Simulation of the model over a segment of test cycle 7 using the reference (current ECU calibration), the optimal, and the causal control solutions. All three cases produce the same cumulative pollutant emissions. The optimal solution saves 2.21% fuel, the causal one 1.91%. Normalised signals are shown for reasons of confidentiality.

The performances of the reference, the optimal solution, and the fully causal control strategy derived from the optimal solution are illustrated in Fig. 5.24. The causal controller is able to closely reproduce the optimal solution. Note that this causal control system is identified by a fully automated procedure requiring only the stationary measurement data for the identification of the engine model.

5.2.4 Optimal NO_x reduction with EGR

Engine B provides several control inputs that affect the fuel consumption but also the NO_x emissions. The EGR valve controls the flow of exhaust gas into the intake manifold. The burnt gas in the intake mixture is inert and reduces the temperature of the combustion zones in the cylinders, which in turn reduces thermal NO_x formation. Conversely, the combustion becomes less efficient. The exhaust flap (EF) can be closed to increase the pressure difference over the EGR valve. A higher EGR mass-flow results, but also higher gas-exchange losses have to be overcome. Finally, a retardation of the injection yields a less efficient combustion and lower NO_x emissions.

Especially during transient operation, it is difficult to derive a control strategy that reduces the NO_x emissions to a desired level while maintaining the lowest possible fuel consumption. For a load increase, the additional constraint of a lower limit on the AFR has to be honoured in order not to produce large peaks of soot emissions. It is shown how a general control strategy can be derived by the analysis of optimal control solutions. The single load step of test case 8 is considered, as shown in the left-hand plot of Fig. 5.25.

The following procedure is applied. First, optimal trajectories for the VGT and the SOI are derived while keeping the EGR valve closed and the EF fully open. This solution is used as reference for a successive reduction of the NO_x emissions by an optimisation of all four control

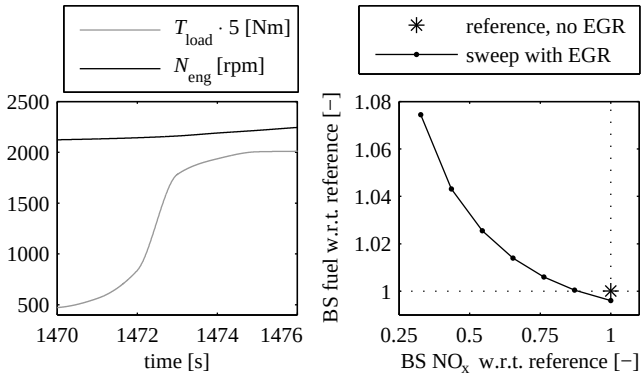

Figure 5.25: The driving profile of test case 8, scaled for engine B (left), and the optimal fuel-NO_x tradeoff for engine B over this cycle (right).

inputs. The result is the optimal fuel-NO_x tradeoff for this load step displayed in the right-hand plot of Fig. 5.25.

Figure 5.26 shows the corresponding control-input trajectories. The following control strategy can be derived. The optimisation does not close the EF at any point. Therefore, this is the least efficient way to reduce the NO_x emissions and should be avoided. During the load increase, the VGT as well as the EGR valve need to be closed such that the AFR stays above the lower limit, which here is chosen at 1.4. A feedforward part based on the gradient of the torque demand could be derived from the solution of the OCP. Finally, the higher NO_x emissions during the load step can be compensated by a transient shift of the SOI and a higher EGR rate during the stationary operation before and after the step.

By extending this case study to a more representative, longer time horizon, sufficient information could be collected to derive an overall controller calibration as presented for engine A in Sec. 5.2.3.

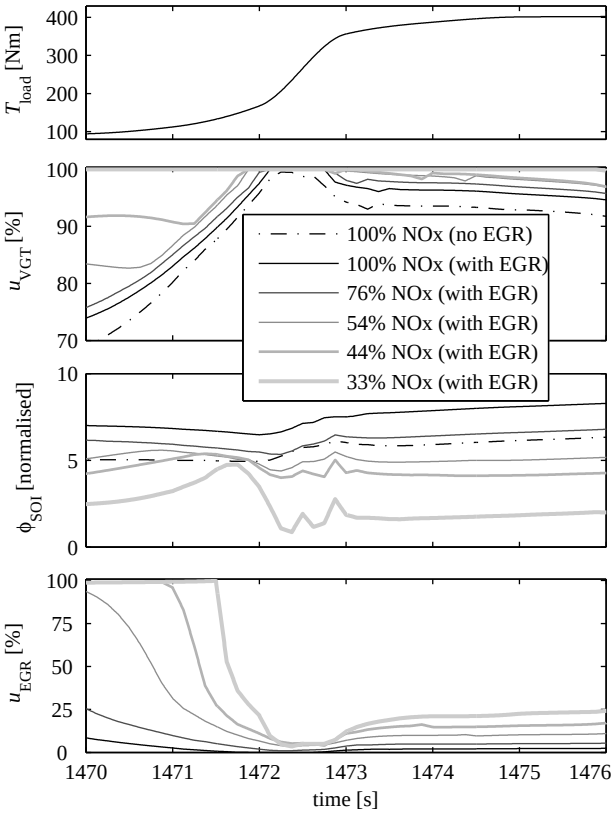

Figure 5.26: Control-input trajectories for the fuel-NO_x tradeoff depicted in the right-hand plot of Fig. 5.25. The EGR valve is closed at 0%. The exhaust flap remains fully open for all cases and thus is not shown. Normalised signals are shown for reasons of confidentiality.

5.3 Experimental validation

This section first describes the test-bench setup required to perform a validation of the dynamic optimisation. Subsequently, the results from the two approaches for the optimal control of diesel engines are presented and discussed.

5.3.1 One-step approach

In order to validate the results from the dynamic optimisation on the engine testbench, three critical points have to be resolved. First, the control signals have to be transmitted to the ECU and the test-bench brake synchronously and sufficiently fast. A master process writes all signals to a shared memory section of the automation system at a frequency of 100 Hz. Using the iLinkRT™ protocol developed by AVL (AVL List GmbH, Graz, Austria) and ETAS (ETAS GmbH, Stuttgart, Germany), the control signals are transferred to an ETAS ES910.3 prototyping and interface module at the same frequency. This module immediately transfers the updated values to the ECU using the ETK interface (ETAS). Simultaneously, an additional process transfers the desired engine speed from the shared memory section to the controller (SPARC by HORIBA Ltd., Kyoto, Japan) of the test-bench dynamometer (HORIBA HD 700 LC). To this end, the proprietary OpenSIM CAN message protocol by HORIBA is used, which ensures a time-synchronous transfer at 100 Hz. These CAN messages, as well as all relevant signals of the ECU are recorded at their natural sampling rate by INCA (ETAS), which runs on a host computer.

The second problem consists of obtaining meaningful and comparable results. When the engine is operated using the ECU, limits such as an operating-point dependent AFR limit are respected. Furthermore, a feedback controller is used to follow the desired load torque. Therefore, the engine does not exactly produce the torque desired by the driving cycle. In addition, the optimal control solution has to provide the same

braking torque during drag phases that results from the current control strategy implemented on the ECU. For these reasons, the effective torque delivered by the engine during a normal run on the test cycle is prescribed during the optimisation.[3] During the validation run, the optimised control inputs are prescribed directly, and no bounds are applied.

Finally, the ECU does not inject exactly the amount of fuel demanded by the corresponding control signal.[4] In fact, the injector maps are identified for a narrow operating region, and the ECU estimation becomes increasingly inaccurate for larger deviations of the combustion control signals from the current calibration. Errors of up to 7% result and thus, the torque produced by the engine is not the same as the desired one. To resolve this mismatch, a correction is applied after the first validation run. The fuel injection is updated according to a Willans approximation [92, Sec. 2.5.1]. The net torque is modelled as a time-variable, affine function of the fuel mass,

$$T_{\text{load}}(t) = \eta_W(t) \cdot m_{\text{fcc}}(t) - T_0(N_{\text{eng}}(t), p_{\text{EM}}(t) - p_{\text{IM}}(t)). \tag{5.5}$$

The loss torque T_0 is the sum of the friction, the inertia, and the gas-exchange work. The former two contributions are estimated using the corresponding submodels of the engine model, whereas the latter is estimated from the pressure difference over the engine measured during the first validation run.

The model is identified using the fuel mass prescribed for the first run, m_{fcc}, and the resulting torque T_{load}. Therefore, the updated amount

[3] This different torque demand causes the change in the predicted fuel savings as compared to the values provided in the description of Fig. 5.24.

[4] The model is identified using data recorded by an external fuel scale.

Table 5.8: Results of the experimental validation. The changes relative to the values measured for the current engine calibration are shown.

test cycle / OCP	*6 / I*	*6 / II*	*7 / I*	*7 / II*
fuel, predicted	−1.28%	−2.85%	−1.52%	−2.49%
measured	−0.85%	−2.39%	−1.42%	−2.50%
NO_x, predicted	0%	25%	0%	10%
measured	−1.05%	21.27%	−3.64%	8.31%

of fuel injected is calculated by

$$m_{\mathrm{fcc,NEW}}(t) = m_{\mathrm{fcc}}(t) \cdot \frac{T_{\mathrm{des}}(t) - T_0(t)}{T_{\mathrm{load}}(t) - T_0(t)}. \tag{5.6}$$

After the correction, the deviation of the torque produced by the engine from the desired load torque hardly ever exceeds 1%.

Results For each of the two test cycles 6 and 7, two OCPs were solved. The first one required the emissions to remain at the level of the current engine calibration (I). The second one allowed a prescribed increase of the NO_x emissions (II). This increase is larger for test cycle 6 since for cycle 7, already high brake-specific (BS) NO_x emissions result from the current calibration. By performing this variation, not only the quantitative accuracy of the model is assessed, but also its ability to reproduce the tradeoff between fuel savings and NO_x emissions is evaluated.

For test cycle 6, the measurement is repeated five times for each set of control inputs to assess its reproducibility. The maximum deviation of any of the five measurements from the average of all of them is used as a measure. For the BS fuel consumption, this deviation is 0.09%, and for the cumulative BS NO_x emissions, the figure is 0.64%. Based on this high reproducibility, the longer test cycle 7 was only measured once for each set of control-input trajectories.

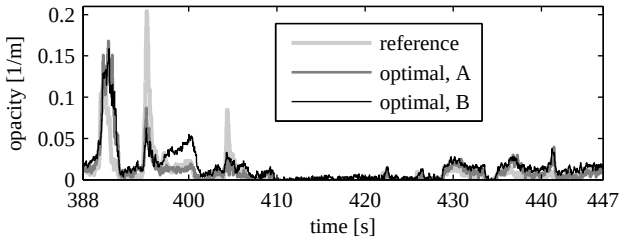

Figure 5.27: Opacity of the exhaust gas measured for the reference and the two optimal solutions. The same segment of test cycle 7 as in Fig. 5.24 is shown.

Table 5.8 summarises the results from the experimental validation. On test cycle 6, the prediction overestimates the fuel savings but manages to predict the NO_x emissions accurately. As can be extracted from the data listed in Table 5.8, the model predicts a factor of 15.9 that relates the relative fuel saving to the corresponding increase of the NO_x emissions when cases I and II are compared. The measured factor is 14.5.

On test cycle 7, the fuel savings are predicted accurately by the model. However, the increase of the NO_x emissions is underestimated. This shift of the model accuracy is due to the fact that the engine is operated in different operating regions on the two cycles. Cycle 6 comprises high-power operating points to the largest extent, whereas cycle 7 prescribes a low-power profile. For cycle 7, the predicted and the measured fuel-to-NO_x factors are 10.3 and 11.1, respectively.

Figure 5.27 shows the opacity of the exhaust gas measured for the reference and the two optimal solutions. As predicted by the model, the overall level remains the same, and thus no active regeneration of the DPF becomes necessary. Furthermore, most instantaneous peaks are even slightly reduced.

5.3.2 Iterative approach

The optimal control trajectories obtained in Sec. 5.2.1 are experimentally validated on the engine testbench. To account for possible deviations in the resulting torque, the emissions and the fuel consumption are related to the integrated engine power. Figure 5.28 summarises the measured cumulative emissions and the fuel consumption.

Five runs are recorded for each set of trajectories. The average as well as the minimum and maximum values are shown in the plots. The emissions remain within the measurement uncertainty, while the fuel consumption is reproducibly reduced. The soot measurement exhibits a large variability, which is also present in the identification data. Therefore, the corresponding model is not reliable and possibly hinders a more effective reduction of the fuel consumption. Furthermore, due to this inconsistency, the model quality seems not to improve when the cross variations are included in the identification data.

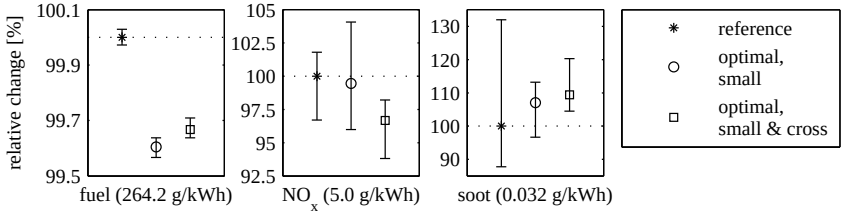

Figure 5.28: Experimental results of the first iteration of the iterative approach to dynamic optimisation. The error bars indicate the range (minimal to maximal values) of the five measurements.

6 Summary and Outlook

For the individual aspects of engine modelling, the numerical methods to solve the optimal control problems for diesel engines, as well as the engineering applications, the conclusions are drawn and the directions to future work are stated in the respective sections. Therefore, a short review of the overall methodology, as well as some concluding remarks and suggestions for further investigations are provided here.

A self-contained set of tools and numerical methods for the efficient solution of optimal control problems for diesel engines is presented. This framework enables the derivation of optimal control-trajectories over long driving profiles. These solutions provide sufficient information to derive complete dynamic engine calibrations. This fully automated model-based approach is illustrated for an engine without EGR. For an engine with EGR, it is shown how optimal control can be utilised to develop a control strategy that provides an optimal transient fuel-NO_x tradeoff. Furthermore, an iterative approach is proposed which combines a transient model refinement with a sequence of related optimal control problems. An experimental validation of the original as

well as of the iterative approach indicates that all findings presented in this thesis accurately transfer to the real engines.

The models used to formulate the optimal control problems are specifically tailored towards this application. They live up to the diverse requirements that enable an efficient solution of an optimisation problem. A high execution speed is unified with the capabilities of smooth interpolation and physically plausible extrapolation. To this end, a physics-based foundation is combined with a setpoint-relative formulation where appropriate. The individual submodels are carefully matched to each other to prevent the propagation and amplification of errors along the model chain.

Further work should focus on a more detailed model for the soot emissions and on the analysis of the singular arcs occurring in the optimal solutions. The time-resolved control inputs could be replaced by the parameters of a predefined control structure in the optimal control problem. The simultaneous nature of the solution process would be preserved, but an optimal controller calibration could be obtained directly. From an engineering point of view, the exhaust-gas aftertreatment system should be included in the model to optimise the interaction between this system and the engine. Finally, advanced model-based control strategies such as model-predictive control could be derived from the modelling approach and the numerical framework presented.

A Nomenclature

The time derivative of a variable x is denoted by $\dot{x}$, whereas $\overset{*}{x}$ represents a flow of mass, heat or energy, for instance. Bold symbols indicate vectors and matrices. The operator ∇ constitutes the gradient of a scalar variable, or the Jacobian matrix of a vector. The Hessian matrix of a scalar variable is indicated by ∇^2. The new value of a variable x in an iterative procedure, as well as its optimal value in the context of optimisation, are denoted by x^*.

The following list introduces the abbreviations and the symbols that are used consistently throughout the text. Indices and specific symbols that are used only in a narrow context are introduced and explained directly at the corresponding locations in the text. For each symbol that can assume different meanings, the respective context is indicated in brackets. Section 4.5 on the iterative solution of the optimal control problem for diesel engines uses a specific nomenclature which is not covered in this list.

A.1 Acronyms and abbreviations

aC	after combustion
AD	algorithmic differentiation
AFM	air-flow meter
AFR	air-to-fuel ratio
af, AFT	adiabatic flame (temperature)
amb	ambient
AS	active set
ATS	(exhaust-gas) after-treatment system
bC	before combustion
BG	burnt gas
BSR	blade-speed ratio
BVP	boundary-value problem
bw	backward
CD	central finite differences
CG	conjugate gradient
CP	compressor
crit	critical
cyl	cylinder
des	desired
DOC	diesel oxydation catalyst
DPF	diesel particulate filter
ECU	engine control unit
EF	exhaust flap
EGR	exhaust-gas recirculation
EM	exhaust manifold
eng	engine
EO	engine-out
ETC	European Transient Cycle
EVO	exhaust valve opening

exh	exhaust
FB, FF	feedback, feedforward
fb	forward-to-backward (ratio)
fcc	fuel (injected) per cylinder and combustion cycle
FD	finite differences
fric	friction
fw	forward
gex	gas exchange
HJB	Hamilton-Jacobi-Bellman
hl	heat losses
HPP	high-pressure pump (common-rail system)
IC	intercooler
ID	ignition delay
IM	intake manifold
IP	interior point
IVC	intake valve closing
KKT	Karush-Kuhn-Tucker (conditions)
LSQ	linear least squares (regression)
mdl	model, modelled
MIMO	multiple-input multiple-output
MSIM	multiple-shooting imitation mode
MV(EM)	mean-value (engine model)
NLP	nonlinear program
ODE	ordinary differential equation
OCP	optimal control problem
OO	operator overloading
opt	optimal
QN	quasi-Newton (update)
QP	quadratic program
reac	reaction
ref	reference

reg	regularisation
RG	residual gas
RH	relative humidity
RHS	right-hand side
rpm	revolutions per minute
sat	saturation, saturated
SCR	selective catalytic reduction
SCT	source-code transformation
SD	symbolic differentiation
SDA	standard dry air
SLQP	sequential linear-quadratic programming
SOC	start of combustion
SOI	start of injection
SP	setpoint
SQP	sequential quadratic programming
surr	surroundings
TB	turbine
TC	turbocharger
TDC	top dead centre
thr	threshold
VGT	variable-geometry turbine
vol	volumetric
WHSC	World-Harmonized Stationary Cycle
WHTC	World-Harmonized Transient Cycle

A.2 Latin symbols

A	*(model)* area
	(optimal control) matrix of Runge-Kutta coefficients
a, b	generic model parameters
B	Hessian approximation
c	*(model)* generic parameter, specific heat, flow coefficient
	(optimal control) path constraint
	(mesh refinement) pre-factor of the truncation error
D	differentiation matrix
e	error of a feedback controller
F	generic objective function
f	*(NO_x model)* reference quantity
	(otherwise) function
g	integrand of a cumulative constraint
G	cumulative value of g
h	*(model)* specific enthalpy
	(transcription) integration step-size
	(nonlinear programming) inequality constraint
H	enthalpy, heating value
J	objective function, i.e. cumulative value of L
k	*(model)* generic parameter
	(mesh refinement) safety margin
	(NO_x model) reaction constant
L	integrand of the objective function
l	linear index
M	*(model, combustion)* molar mass
	(optimal control) number of discretisation points
m	*(model)* mass
	(time subdivision of OCPs) number of intervals
n	number

N	*(model)* rotational speed
	(optimal control) number of time steps
p	*(model)* pressure
	(SQP) Newton step
	(mesh refinement) step-refinement factor
Q	heat
R	specific gas constant
r	reaction rate
s	*(optimal control)* collocation order
	(regularisation) slope
T	*(model)* torque
	(optimal control) time horizon
t	time
u	control input
V	volume
W, w	global and interval-specific quadrature weights
x	state variable, fraction
y	generic model output

A.3 Greek symbols

α, β	composition of fuel $C_\alpha H_\beta$
γ	polytropic coefficient
Δ	difference
δ	infinitesimal difference
ε	tolerance, precision
ζ	molar or volumetric fraction
η	efficiency
Θ	rotational inertia
ϑ	temperature
κ	isentropic coefficient
λ	*(model)* air-to-fuel ratio
	(optimisation) Lagrange multiplier
μ	Lagrange multiplier
ξ	mass fraction
Π	pressure ratio
π	time-variable parameter
ρ	density
τ	*(model)* characteristic time
	(nonlinear programming) barrier parameter
	(mesh refinement) truncation error
	(time subdivision of OCPs) interval boundary
φ	angle
Ψ	flow function
ω	*(model)* rotational speed
	(optimisation) NLP variable

B Combustion

Four species are used to represent all mixtures in the model, namely oxygen O_2, nitrogen N_2, water H_2O, and carbon dioxide CO_2. Their molar masses are (32.00, 28.01, 18.02, 44.01) g/mol, respectively. Fuel is represented by $C_\alpha H_\beta$ with the molar mass $M_{\text{fuel}} = 12\alpha + \beta$. The combustion of one fuel molecule is described by the reaction

$$C_\alpha H_\beta + \left(\alpha + \frac{\beta}{4}\right) O_2 \longrightarrow \alpha CO_2 + \frac{\beta}{2} H_2O. \tag{B.1}$$

The combustion of one fuel molecule thus introduces or removes, as the case may be, the following number of molecules of the four species constituting the gas.

$$\Delta \bar{\boldsymbol{n}} = \begin{pmatrix} -\left(\alpha + \frac{\beta}{4}\right) & 0 & \alpha & \frac{\beta}{2} \end{pmatrix}^T. \tag{B.2}$$

The molar amount of each species j in the cylinder before the combustion is

$$n_{\text{bC},j} = \frac{(1 - x_{\text{RG}}) \cdot m_{\text{cyl}} \cdot \xi_{\text{in},j} + x_{\text{RG}} \cdot m_{\text{cyl}} \cdot \xi_{\text{aC},j}}{M_j}. \tag{B.3}$$

The subscript "in" denotes the freshly aspirated mixture, which has the same composition as the gas in the intake manifold. After the combustion, the amount of species j becomes

$$n_{\mathrm{aC},j} = n_{\mathrm{bC},j} + \Delta\bar{n}_j \cdot n_{\mathrm{fuel}} = n_{\mathrm{bC},j} + \Delta\bar{n}_j \cdot \frac{m_{\mathrm{fcc}}}{M_{\mathrm{fuel}}}. \tag{B.4}$$

Converting back to a mass fraction and inserting Eq. (B.3) yields

$$\begin{aligned}\xi_{\mathrm{aC},j} &= \frac{n_{\mathrm{aC},j} \cdot M_j}{m_{\mathrm{cyl}} + m_{\mathrm{fcc}}} \\ &= \frac{(1 - x_{\mathrm{RG}}) \cdot m_{\mathrm{cyl}} \cdot \xi_{\mathrm{in},j} + x_{\mathrm{RG}} \cdot m_{\mathrm{cyl}} \cdot \xi_{\mathrm{aC},j} + m_{\mathrm{fcc}} \cdot \frac{M_j}{M_{\mathrm{fuel}}} \cdot \Delta\bar{n}_j}{m_{\mathrm{cyl}} + m_{\mathrm{fcc}}}.\end{aligned} \tag{B.5}$$

This equation can be solved for $\xi_{\mathrm{aC},j}$, resulting in

$$\xi_{\mathrm{aC},j} = \frac{(1 - x_{\mathrm{RG}}) \cdot m_{\mathrm{cyl}} \cdot \xi_{\mathrm{in},j} + m_{\mathrm{fcc}} \cdot \frac{M_j}{M_{\mathrm{fuel}}} \cdot \Delta\bar{n}_j}{(1 - x_{\mathrm{RG}}) \cdot m_{\mathrm{cyl}} + m_{\mathrm{fcc}}}. \tag{B.6}$$

The gas composition before combustion is

$$\boldsymbol{\xi}_{\mathrm{bC}} = (1 - x_{\mathrm{RG}}) \cdot \boldsymbol{\xi}_{\mathrm{in}} + x_{\mathrm{RG}} \cdot \boldsymbol{\xi}_{\mathrm{aC}}. \tag{B.7}$$

Equation (B.1) can be normalised by α. As a result, the only parameter of the fuel relevant for the mass fractions in the gases is its hydrogen-to-carbon ratio β/α. In the work at hand, this ratio was chosen as 1.9535.

B.1 Composition of air and burnt gas

Standard dry air (SDA) is usually assumed to contain 20.95% oxygen, 0.93% argon, and 0.03% carbon dioxide, in measures of mole fractions. Here, argon and all trace substances are attributed to nitrogen, i.e.

$$\begin{aligned} \zeta_{\mathrm{SDA,O_2}} &= 0.2095, \\ \zeta_{\mathrm{SDA,CO_2}} &= 0.0003, \\ \zeta_{\mathrm{SDA,N_2}} &= 1 - \zeta_{\mathrm{DA,O_2}} - \zeta_{\mathrm{DA,N_2}} = 0.7902. \end{aligned} \tag{B.8}$$

From the measured relative humidity, the ambient temperature and the barometric pressure, the molar fraction of water can be calculated. The relative humidity is defined as the ratio of the partial pressure of water to its saturated vapour pressure,

$$\mathrm{RH} = \frac{p_{\mathrm{H_2O}}}{p_{\mathrm{H_2O,sat}}} = \frac{\zeta_{\mathrm{air,H_2O}} \cdot p_{\mathrm{amb}}}{p_{\mathrm{H_2O,sat}}}. \tag{B.9}$$

The saturated vapour pressure is calculated by Wagner's equation [178]

$$\begin{aligned} \ln\left(\frac{p_{\mathrm{H_2O,sat}}}{p_{\mathrm{H_2O,crit}}}\right) = \frac{\vartheta_{\mathrm{crit}}}{\vartheta_{\mathrm{amb}}} \cdot \big(& a_1 \cdot \tau_{\mathrm{amb}} + a_2 \cdot \tau_{\mathrm{amb}}^{1.5} + a_3 \cdot \tau_{\mathrm{amb}}^{3} + \\ & a_4 \cdot \tau_{\mathrm{amb}}^{3.5} + a_5 \cdot \tau_{\mathrm{amb}}^{4} + a_6 \cdot \tau_{\mathrm{amb}}^{7.5} \big), \end{aligned} \tag{B.10}$$

with the parameters

$$\begin{aligned} p_{\mathrm{H_2O,crit}} &= 220.64 \text{ bar}, & \vartheta_{\mathrm{crit}} &= 647.096 \text{ K}, \\ a_0 &= -7.85951783, & a_2 &= 1.84408259, \\ a_3 &= -11.7866497, & a_4 &= 22.6807411, \\ a_5 &= -15.9618719, & a_6 &= 1.80122502. \end{aligned}$$

Once the molar fraction of water in the (wet) air is known, the fractions of the other species are calculated as

$$\zeta_{\mathrm{air},j} = (1 - \zeta_{\mathrm{air,H_2O}}) \cdot \zeta_{\mathrm{SDA},j}. \tag{B.11}$$

Finally, the molar mass of wet air and the mass fractions of all species are obtained,

$$M_{\text{air}} = \sum_j \zeta_{\text{air},j} \cdot M_j, \quad \xi_{\text{air},j} = \zeta_{\text{air},j} \cdot \frac{M_j}{M_{\text{air}}}. \tag{B.12}$$

Burnt gas is defined as the mixture resulting from a stoichiometric combustion with air. Therefore, its composition only depends on the fuel and the composition of the air. By setting $\xi_{\text{in}} = \xi_{\text{air}}$, $x_{\text{RG}} = 0$ and demanding $\xi_{\text{aC,O}_2} = 0$, Eq. (B.6) for species O_2 can be solved for

$$m_{\text{cyl,stoich}} = \frac{m_{\text{fcc}} \cdot M_{\text{O}_2} \cdot \delta}{M_{\text{fuel}} \cdot \xi_{\text{air,O}_2}}, \text{ with } \delta := \alpha + \frac{\beta}{4}. \tag{B.13}$$

When this expression is inserted in Eq. (B.6), the fuel mass cancels and the composition of the burnt gas is obtained as

$$\xi_{\text{BG},j} = \frac{M_{\text{O}_2} \cdot \delta \cdot \xi_{\text{air},j} + M_j \cdot \Delta\bar{n}_j \cdot \xi_{\text{air,O}_2}}{M_{\text{O}_2} \cdot \delta + M_{\text{fuel}} \cdot \xi_{\text{air,O}_2}}. \tag{B.14}$$

B.2 Adiabatic flame temperature

For a detailed description of adiabatic combustion, see [102]. Assume a global single-directed reaction from reactants R_j to products P_j involving S species

$$\sum_{j=1}^{S} \nu'_j \mathrm{R}_j \longrightarrow \sum_{j=1}^{S} \nu''_j \mathrm{P}_j. \tag{B.15}$$

The reactants are at a known temperature ϑ_r. By the heat of reaction, the products are heated to the unknown temperature ϑ_p, which is the adiabatic flame temperature (AFT) for combustion reactions. For diesel combustion, constant pressure and therefore conservation of enthalpy is presumed. Furthermore, complete and stoichiometric combustion is assumed and dissociations are neglected. The molar engineering enthalpy h^M consists of the formation and sensible contributions

$$h^\mathrm{M}(\vartheta) = \Delta H_\mathrm{f}^{\mathrm{o,M}} + h_\mathrm{s}^\mathrm{M}(\vartheta) = \sum_{j=1}^{S} \zeta_j \left(\Delta H_{\mathrm{f},j}^{\mathrm{o,M}} + \int_{\vartheta_0}^{\vartheta} c_{\mathrm{p},j}^\mathrm{M}(\tau)\, d\tau \right), \tag{B.16}$$

where ζ_j is the molar fraction of species j in the mixture. Separating the enthalpy balance for reaction (B.15) into formation and sensible enthalpies yields

$$\begin{aligned} \sum_{j=1}^{S} \nu''_j h_{\mathrm{s},j}^\mathrm{M}(\vartheta_p) - \sum_{j=1}^{S} \nu'_j h_{\mathrm{s},j}^\mathrm{M}(\vartheta_r) &= \sum_{j=1}^{S} (\nu'_j - \nu''_j) \Delta H_{\mathrm{f},j}^{\mathrm{o,M}} \\ &= Q_\mathrm{fuel}^\mathrm{M} = Q_\mathrm{fuel} \cdot M_\mathrm{fuel}, \end{aligned} \tag{B.17}$$

which states that the difference in sensible enthalpies between products and reactants has to equal the (molar) heat of reaction. The latter is a property of the fuel and is commonly specified as the mass-specific heating value Q_fuel. Presuming no condensation of the products, the

lower heating value is approximated as $H_\mathrm{l} = 42^{\mathrm{MJ/kg}}$ for diesel fuel. Note that with this formulation, the stoichiometric coefficients ν_j have to be normalised such that one mole of fuel is consumed during the reaction,

$$\begin{aligned} &\mathrm{C}_\alpha\mathrm{H}_\beta + \delta\ \mathrm{O}_2 + \delta\cdot r_{\mathrm{N}_2}\ \mathrm{N}_2 + \delta\cdot r_{\mathrm{H}_2\mathrm{O}}\ \mathrm{H}_2\mathrm{O} + \delta\cdot r_{\mathrm{CO}_2}\ \mathrm{CO}_2 \\ &\quad\longrightarrow\ \delta\cdot r_{\mathrm{N}_2}\ \mathrm{N}_2 + \left(\frac{\beta}{2} + \delta\cdot r_{\mathrm{H}_2\mathrm{O}}\right)\mathrm{H}_2\mathrm{O} + (\alpha + \delta\cdot r_{\mathrm{CO}_2})\mathrm{CO}_2, \end{aligned} \tag{B.18}$$

where $r_j := \zeta_j/\zeta_{\mathrm{O}_2}$ is the molar fraction of species j in the cylinder before combustion, related to the one of oxygen.

The enthalpy balance (B.17) has to be solved for ϑ_p, which is usually done by iteration using lookup tables or polynomial approximations for the enthalpies. Here, the sixth-order polynomials from [46] are used. An iterative calculation has to be avoided for two reasons. First, the execution speed of the model would be substantially reduced by introducing an iteration in the model function. Second, since an iteration with a certain termination criterion is an adaptive algorithm, it has to be considered an inconsistent function generator, which is detrimental in the context of optimisation.

In order to avoid an iterative calculation, the temperature dependencies are linearised in the two relevant temperature regions 800–1000K for the reactants at SOC and 1800–2600K for the products at the AFT, i.e.

$$h^M_{s,j}(\vartheta) \approx \begin{cases} h^r_{0,j} + \vartheta\cdot h^r_{1,j}, & \text{if } \vartheta \in [800, 1000], \\ h^p_{0,j} + \vartheta\cdot h^p_{1,j}, & \text{if } \vartheta \in [1800, 2600]. \end{cases} \tag{B.19}$$

For each of the two relevant intervals and for each species j, the sum of squared errors of the linearisation as compared to the original polynomial is minimised over the entire interval, yielding the two coefficients. The error introduced by the linearisation is below 0.4% throughout the

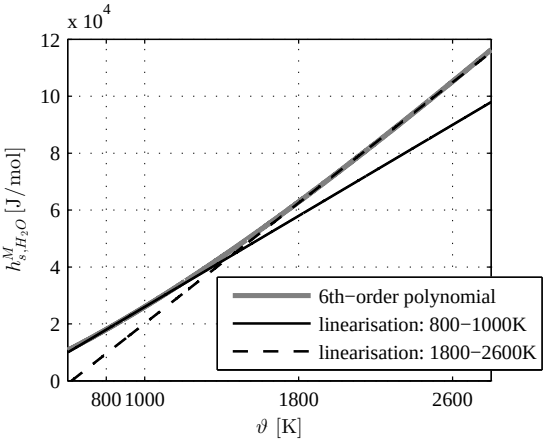

Figure B.1: Linearisation of the sensible enthalpy of water in the two temperature ranges relevant for the calculation of the adiabatic flame temperature.

two relevant intervals. As an example, Fig. B.1 displays the original polynomial and the two linearisations for the species H_2O.

Fuel is represented by 1-octene C_8H_{16} in terms of thermodynamic properties, except for its heating value. The sensible enthalpy of the reactants and products is thus given by

$$h_{\mathrm{s}}^{\mathrm{M}}(\vartheta) = \underbrace{\sum_j a_{j,0} \cdot \nu_j}_{h_{\mathrm{s},0}^{\mathrm{M}}} + \vartheta \cdot \underbrace{\sum_j a_{j,1} \cdot \nu_j}_{h_{\mathrm{s},1}^{\mathrm{M}}}, \tag{B.20}$$

with $j = \{O_2,\ N_2,\ H_2O,\ CO_2,\ C_8H_{16}\}$. Using this notation, the explicit formulation of the AFT finally reads

$$\vartheta_{\mathrm{af}} = \frac{h_{\mathrm{s},0}^{\mathrm{r,M}} + \vartheta_{\mathrm{SOC}} \cdot h_{\mathrm{s},1}^{\mathrm{r,M}} - h_{\mathrm{s},0}^{\mathrm{p,M}}}{h_{\mathrm{s},1}^{\mathrm{p,M}}}. \tag{B.21}$$

C Processing of measurement data

The raw data provided by the measurement devices need to be processed before they can be used for the extraction of relevant information or directly for the identification and validation of models. First, the filtering and post-processing of the in-cylinder pressure signal is presented, followed by the description of the procedure to calculate the cylinder charge and its composition from the measured signals. Finally, it is shown how the information on the combustion is extracted from the in-cylinder pressure trace.

C.1 Cylinder pressure

The cylinder-pressure data is stored in AVL's proprietary I-file format. The open-source software "catool" (http://www.catool.org/) is used to convert it to MATLAB data. This tool is called by a batch file which runs through all measurement points in a measurement series and generates a temporary configuration file. The analysis functionalities of the tool are not used.

Three main steps are required to attain a cylinder-pressure signal which provides physically meaningful information, namely filtering, compensation for the thermodynamic top-dead centre (TDC), and pressure pegging. The filtering eliminates noise, indicating-bore oscillations, and cylinder resonance effects. Since the thermodynamic TDC does not coincide with the mechanical one, this effect has to be accounted for to obtain consistent information on the combustion. Finally, the pressure pegging identifies the offset of the relative pressure sensors. All three steps are described in the following sections.

C.1.1 Filtering

Since cycle-to-cycle variations are not taken into account, 50 cycles are averaged for all stationary measurements. Two approaches for the further filtering of the averaged signal are considered. The first focuses on indicating-bore oscillations, whereas the second applies a general low-pass filter.

The first approach is described in [144]. In case the pressure sensor is not installed flush with the combustion-chamber wall, the indicating bore introduces a cavity adjoint to the combustion chamber. Depending on the geometry of this bore and the distance of the pressure sensor from the combustion chamber, a different resonance frequency results. The paper cited approximates the cavity of the indicating bore as a Helmholtz resonator. This model is inverted and the corresponding fre-

quency spectrum is subtracted from the cylinder-pressure signal. This method was used to process the data of engine A, for which a technical drawing of the sensor-adapter setup was available.

The second approach is described in [150]. This approach does not rely on geometrical data, but relies on empirical rules or an ad-hoc analysis of the frequency spectrum. A perfect low-pass filter is applied to the signal in the frequency domain. A smooth roll-off of the pass band is used to suppress oscillations in the time domain. The cutoff harmonic and the width of the roll-off are either chosen as an empirically identified function of the engine speed and the relative load, by analysing the characteristics of the frequency spectra for motored and fired operation, or by tuning the parameters by hand.

The magnitude of the frequency spectrum at harmonic k is multiplied by

$$\theta_k := \begin{cases} 1, & \text{if } k < k_\mathrm{c} - k_\mathrm{stop}/2, \\ 0, & \text{if } k > k_\mathrm{c} + k_\mathrm{stop}/2, \\ \frac{1}{2} \cdot \left[\cos\left(\frac{k - k_\mathrm{c} + k_\mathrm{stop}/2}{k_\mathrm{stop}} \cdot \pi \right) + 1 \right], & \text{otherwise.} \end{cases} \tag{C.1}$$

Figure C.1 shows this filter for the parameter values $\{k_\mathrm{c}, k_\mathrm{stop}\} = \{150, 120\}$, which is an arbitrary choice to illustrate the general shape of the filter.

The rules determining the filter parameters, which are empirically derived in the original literature, are described next. At no load (i.e. motored operation), the cutoff harmonic and the width of the roll-off are

$$\{\underline{k}_\mathrm{c}, \underline{k}_\mathrm{stop}\} = \{55, 65\}. \tag{C.2a}$$

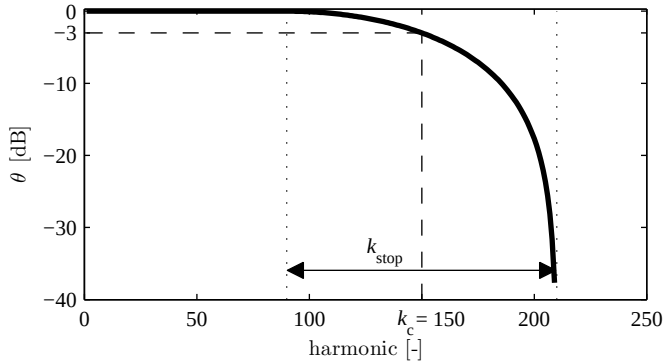

Figure C.1: Kernel of the lowpass filter for the cylinder pressure as proposed in [150].

At full load, the two parameters depend on the engine speed and are described by

$$\{\overline{k}_\mathrm{c}, \overline{k}_\mathrm{stop}\} = \begin{cases} \{240, 300\}, & \text{if } N_\mathrm{eng} < 1250\text{rpm}, \\ \{144, 204\}, & \text{if } N_\mathrm{eng} > 4000\text{rpm}, \\ \{283.625 - 0.0349 \cdot N_\mathrm{eng}, \bar{k}_\mathrm{c} + 60\}, & \text{otherwise.} \end{cases} \tag{C.2b}$$

The actually used values are found by linear interpolation, with the relative load RL as argument,

$$k_\bullet = (1 - RL) \cdot \underline{k}_\bullet + RL) \cdot \overline{k}_\bullet, \tag{C.2c}$$

where

$$RL = \frac{m_\mathrm{fcc}}{m_\mathrm{fcc,max}(N_\mathrm{eng})}. \tag{C.2d}$$

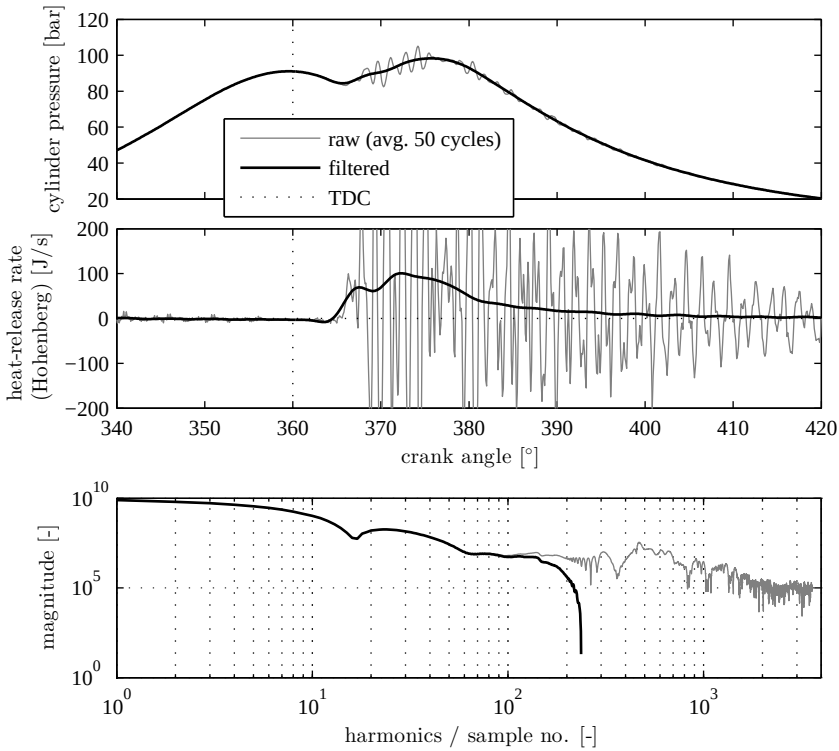

Figure C.2: Result of the cylinder-pressure filtering. Engine B, 2300 rpm and 260 Nm. The values of the filter parameters used are calculated by (C.2), yielding $k_c = 144.8$ and $k_{stop} = 185.1$.

The time-domain signals resulting from the filtering procedure just described closely resemble smoothing splines fitted to the data. An example is provided in Fig. C.2. An operating point of medium speed and medium load of engine B is shown. Besides the time and frequency signals, also the apparent heat-release rate calculated from the cylinder pressure is provided to highlight the importance of the filtering. The heat-release rate is calculated form the cylinder pressure by [78, 102, Sec. 10.4]

$$\frac{dQ_{\mathrm{app}}}{d\varphi} = \frac{\kappa}{\kappa-1} \cdot p_{\mathrm{cyl}} \cdot \frac{dV_{\mathrm{cyl}}}{d\varphi} + \frac{1}{\kappa-1} \cdot V_{\mathrm{cyl}} \cdot \frac{dp_{\mathrm{cyl}}}{d\varphi}. \tag{C.3}$$

For the plot shown, $\kappa = 1.35$ is used.

This second approach for the cylinder-pressure filtering does not require any knowledge of the combustion-chamber geometry or the pressure-sensor setup. The two filter parameters easily can be adjusted by hand based on the resulting time-domain pressure signal or the corresponding heat-release rate. Alternatively, or as a starting point, the empirical correlations described above provide dependable values.

C.1.2 Thermodynamic top dead centre

The thermodynamic TDC does not coincide with the mechanical one. A wrong apparent TDC disturbs the pressure pegging described in the next section. The mismatch is identified using motored measurements throughout the speed range of the engine under consideration. The filtered cylinder-pressure signal is interpolated by cubic splines between the samples, allowing to accurately obtain the location of maximum pressure. Figure C.3 shows the resulting data plotted over the engine speed.

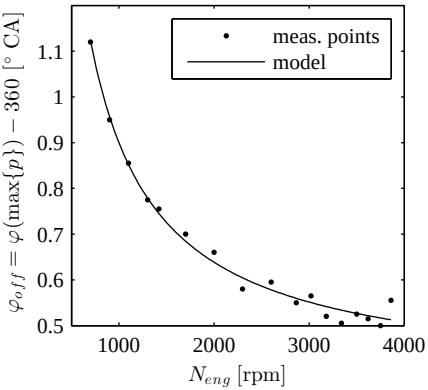

Figure C.3: Data and model for the difference between the mechanical and the thermodynamic top dead centre, engine B.

A model that is able to closely reproduce the observed trend is

$$\varphi_{\text{off}} = k_{\varphi,0} + k_{\varphi,1} \cdot \exp\left(\frac{1}{N_{\text{eng}}}\right). \tag{C.4}$$

A linear least-squares regression is used to identify the two parameters of this model. The result is shown in Fig. C.3. The data is accurately reproduced by the model.

C.1.3 Pressure pegging

The piezoresistive pressure sensors used for the measurement of the in-cylinder pressure are subject to drift caused by thermal stress. Thus, they only provide relative pressure information. To obtain the absolute pressure level, several methodologies are available. A fast measurement of the pressure in the intake manifold, close to the intake valves, was not available. Thus, the mean pressure in the intake manifold was used to estimate the pressure offset. As a second step, the polytropic relation is used to match the compression [123].

The objective is to minimise the square of the difference between the measured and the estimated pressure,

$$\min_{p_{\text{offs}}} \left\{ \int_{\varphi_{\text{IVC}}}^{\varphi_{\text{SOI}}} \left[(p_{\text{cyl,ms}}(\varphi) - p_{\text{offs}}) - p_{\text{cyl,est}}(\varphi) \right]^2 \, d\varphi \right\}. \tag{C.5}$$

To calculate the pressure $p_{\text{cyl,est}}$, two approaches are applicable. The first is to assume isentropic compression, and to calculate the isentropic coefficient from the gas composition and the temperature. To obtain the initial temperature, the cylinder charge has to be known, which is described in the next section. Starting at $\hat{p}(\varphi_{\text{IVC}})$ and ϑ_{IVC}, the isentropic compression of an ideal gas is calculated incrementally by

$$\frac{dp}{dV} = -\kappa(\vartheta(p,V)) \cdot \frac{p}{V}. \tag{C.6}$$

A forward Euler integration with stepsize $\Delta\varphi = 0.1°$ is applied, yielding

$$p_{\text{cyl,is}}(\varphi+\Delta\varphi) = p_{\text{cyl,is}}(\varphi) \cdot \left(1 + \frac{V_{\text{cyl}}(\varphi) - V_{\text{cyl}}(\varphi+\Delta\varphi)}{V_{\text{cyl}}(\varphi)} \cdot \kappa_{\text{cyl}}(\vartheta_{\text{cyl,is}}(\varphi)) \right). \tag{C.7}$$

The temperature is defined by the ideal-gas law, and the temperature-dependent isentropic coefficient $\kappa_{\text{cyl}} = c_{\text{p}}/c_{\text{v}}$ is calculated using the polynomial representation of the thermodynamic properties as described in Appendix B.

Due to heat transfers between the cylinder charge and the cylinder walls, assuming isentropic compression introduces certain errors. Either, a model for the heat transfers can be used, which however has to be identified for any given engine geometry. Alternatively, polytropic compression can be assumed, which is described next.

The second approach assumes a constant polytropic coefficient γ for each measurement point. This additional parameter is identified along with the pressure offset, i.e. it is a second free variable when solving (C.5), and

$$p_{\mathrm{cyl,est}}(\varphi) = p_{\mathrm{IVC}} \cdot \left(\frac{V_{\mathrm{IVC}}}{V(\varphi)} \right)^{\gamma} . \tag{C.8}$$

The advantage of this approach is that heat transfers between gas and walls are inherently accounted for and do not disturb the pressure pegging. In addition, the cylinder charge and its composition do not have to be known in advance. The latter fact enables a separate calculation of the absolute pressure level and the cylinder charge, which reduces the probability of physically wrong cross corrections between the parameters to be identified.

The maximum pressure difference between the measured and the polytropic pressure traces typically is in the range of 0.03% and hardly exceeds 0.1% for any measurement considered. To check the results for outliers and plausibility, maps of the pressure offset over the intake-manifold pressure and the polytropic coefficient identified at different operating conditions can be considered. Figure C.4 shows an example for engine B. The engine map without EGR is shown.

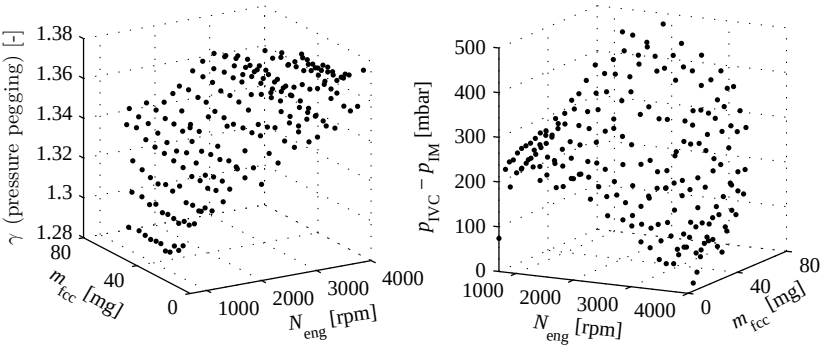

Figure C.4: Maps resulting from the cylinder-pressure pegging, engine B without EGR.

C.2 Cylinder charge and composition

Once the in-cylinder pressure is available, the cylinder charge and its composition can be calculated. The air mass entering each cylinder is calculated from the flow measured by the air-flow meter (AFM),

$$m_{\text{air}} = \frac{\overset{*}{m}_{\text{AFM}} \cdot 120}{N_{\text{eng}} \cdot n_{\text{cyl}}}. \tag{C.9}$$

Similarly, the EGR mass m_{EGR} that enters each individual cylinder can be calculated from the EGR mass-flow $\overset{*}{m}_{\text{EGR}}$. The latter is estimated as described in the next section. The total mass of fresh gas entering each cylinder is

$$m_{\text{in}} = m_{\text{air}} + m_{\text{EGR}}, \tag{C.10}$$

where m_{EGR} is set to zero if the engine does not have an external EGR system.

The temperatures of the fresh gas and the residual gas (RG) are estimated by the isentropic relation,

$$\vartheta_{\text{in}} = \vartheta_{\text{IM}} \cdot \left(\frac{p_{\text{IM}}}{\hat{p}(\varphi_{\text{IVC}})} \right)^{\frac{1-\kappa_{\text{in}}}{\kappa_{\text{in}}}}, \quad \vartheta_{\text{RG}} = \vartheta_{\text{EM}} \cdot \left(\frac{p_{\text{EM}}}{\hat{p}(\varphi_{\text{IVC}})} \right)^{\frac{1-\kappa_{\text{RG}}}{\kappa_{\text{RG}}}}. \tag{C.11}$$

Due to the small ratios between the pressures in the manifolds and the pressure in the cylinder at IVC, the choice of the isentropic coefficients is not critical. Values of $\kappa_{\text{in}} = 1.4$ and $\kappa_{\text{RG}} = 1.35$ are used here. A change of 0.1 in either of the two and in both directions changes the calculated values of the residual-gas fraction x_{RG} and the total cylinder mass m_{cyl} by less than 0.3%.

The following iterative scheme is used to calculate the cylinder charge and its composition.

1. Initialise $R_{\mathrm{IVC}} = 284$.

2. Evaluate a thermal balance to calculate the residual-gas fraction,

$$x_{\mathrm{RG}} = \frac{\frac{p_{\mathrm{IVC}} \cdot V_{\mathrm{IVC}}}{R_{\mathrm{IVC}} \cdot m_{\mathrm{in}}} - \vartheta_{\mathrm{in}}}{\vartheta_{\mathrm{RG}} - \vartheta_{\mathrm{in}} + \frac{p_{\mathrm{IVC}} \cdot V_{\mathrm{IVC}}}{R_{\mathrm{IVC}} \cdot m_{\mathrm{in}}}}. \tag{C.12a}$$

3. Calculate the temperature as the mass-weighted average

$$\vartheta_{\mathrm{IVC}} = x_{\mathrm{RG}} \cdot \vartheta_{\mathrm{RG}} + (1 - x_{\mathrm{RG}}) \cdot \vartheta_{\mathrm{in}} \tag{C.12b}$$

and the cylinder charge from the ideal gas law,

$$m_{\mathrm{cyl}} = \frac{p_{\mathrm{IVC}} \cdot V_{\mathrm{IVC}}}{R_{\mathrm{IVC}} \cdot \vartheta_{\mathrm{IVC}}}. \tag{C.12c}$$

4. The total exhaust-gas fraction is composed of the (internal) residual gas and the externally recirculated exhaust gas,

$$x_{\mathrm{RG,tot}} = \frac{m_{\mathrm{RG}} + m_{\mathrm{EGR}}}{m_{\mathrm{cyl}}} = x_{\mathrm{RG}} + \frac{m_{\mathrm{EGR}}}{m_{\mathrm{cyl}}}. \tag{C.12d}$$

Note that if no external EGR is present, $x_{\mathrm{RG,tot}} = x_{\mathrm{RG}}$. The mass fractions of all species in the gas after and before combustion can now be calculated by adapting Eqs. (B.6) and (B.7),

$$\xi_{\mathrm{aC},j} = \frac{(1 - x_{\mathrm{RG,tot}}) \cdot m_{\mathrm{cyl}} \cdot \xi_{\mathrm{air},j} + m_{\mathrm{fcc}} \cdot \frac{M_j}{M_{\mathrm{fuel}}} \cdot \Delta \bar{n}_j}{(1 - x_{\mathrm{RG,tot}}) \cdot m_{\mathrm{cyl}} + m_{\mathrm{fcc}}}, \tag{C.12e}$$

$$\boldsymbol{\xi}_{\mathrm{bC}} = (1 - x_{\mathrm{RG,tot}}) \cdot \boldsymbol{\xi}_{\mathrm{air}} + x_{\mathrm{RG,tot}} \cdot \boldsymbol{\xi}_{\mathrm{aC}}. \tag{C.12f}$$

5. Finally, the gas constant of the cylinder charge is the mass-weighted average of its species,

$$R_{\mathrm{IVC}} = \boldsymbol{\xi}_{\mathrm{bC}}^{T} \cdot \boldsymbol{R}. \tag{C.12g}$$

6. Check if the relative change, as compared to the last iteration, of x_{RG}, ϑ_{IVC}, and m_{cyl} is below a prescribed tolerance. If not, repeat steps 2 to 5 and check the relative change again.

Besides the termination tolerance which is chosen as 10^{-3}, a maximum number of iterations of 20 is prescribed. However, for the data at hand, three iterations were always sufficient to achieve the requested tolerance.

C.2.1 Calculation of the EGR mass-flow

The various definitions of EGR rates and different ways of calculating them are summarised in [138]. The most reliable and accurate approach is to measure the concentrations of any gas species in the exhaust and intake manifolds. These signals are then used to solve a balance equation for the desired quantity, e.g. the EGR ratio, the EGR mass in the intake manifold, or the EGR mass-flow. If no such special measurement equipment is available, two options remain.

Thermal balance For uncooled EGR, a thermal balance can be applied to calculate the fraction of exhaust gas x_{EG} in the intake manifold [134]. The equation

$$c_{\mathrm{v,IM}} \cdot \vartheta_{\mathrm{IM}} = x_{\mathrm{EG}} \cdot c_{\mathrm{v,EG}} \cdot \vartheta_{\mathrm{EG}} + (1 - x_{\mathrm{EG}}) \cdot c_{\mathrm{v,air}} \cdot \vartheta_{\mathrm{air}} \tag{C.13}$$

needs to be solved for x_{EG}, which necessitates an iterative solution since the specific heat c_{v} depends on the gas composition. However,

assuming a constant value of c_v allows the explicit solution to be written down as

$$x_{EG} = \frac{\vartheta_{IM} - \vartheta_{air}}{\vartheta_{EG} - \vartheta_{air}}. \tag{C.14}$$

Compared to the other uncertainties such as heat transfers in the manifolds and the EGR piping, the error introduced by this simplification is negligible.

For engines with cooled EGR, the temperature difference between the cooled exhaust gas and the mixture in the intake manifold is too small for a reliable calculation of x_{EG}. In this case, only the approach described next can be applied.

Mass balance The notion is to identify the volumetric efficiency and thus the mass flow into the cylinders using measurements without EGR. This model is then used to derive an estimate $\overset{*}{m}_{cyl}$ of the cylinder mass-flow for the measurements with EGR. Using this estimate, the EGR mass-flow can be calculated by

$$\overset{*}{m}_{EGR} = \overset{*}{m}_{cyl} - \overset{*}{m}_{air}. \tag{C.15}$$

The air mass-flow thereby is measured by the AFM.

The cylinder mass-flow is roughly proportional to the density in the intake manifold. However, due to internal EGR (i.e. residual exhaust gas remaining in the cylinders) and valve-opening overlap, the pressure difference accross the engine and the engine speed do have an effect as

well. For the EGR-equipped version of engine B, the model

$$\frac{\overset{*}{m}_{\mathrm{cyl}} \cdot 120}{N_{\mathrm{eng}} \cdot n_{\mathrm{cyl}}} = m_{\mathrm{cyl}} = k_0 + k_1 \cdot \frac{p_{\mathrm{IM}}}{\vartheta_{\mathrm{IM}}} + k_2 \cdot N_{\mathrm{eng}} + k_3 \cdot N_{\mathrm{eng}}^2 + k_4 \cdot (p_{\mathrm{EM}} - p_{\mathrm{IM}}) + k_5 \cdot (p_{\mathrm{EM}} - p_{\mathrm{IM}}) \cdot N_{\mathrm{eng}} \quad \text{(C.16)}$$

is found to be able to accurately reproduce the volumetric efficiency. For the identification, the stationary engine map as well as the variations of the VGT and the exhaust flap are used. This data ensures a reliable identification of the influences of the pressure difference and the density in the intake manifold.

Figure C.5 shows the mass that enters each cylinder as a function of the engine speed and the density in the intake manifold. Note that the curvature in the direction of the engine speed cannot be perceived in this view. However, the model quality decreases considerably when any of the two engine-speed terms in (C.16) are omitted. Figure C.6 shows the measured cylinder mass versus the model fit.

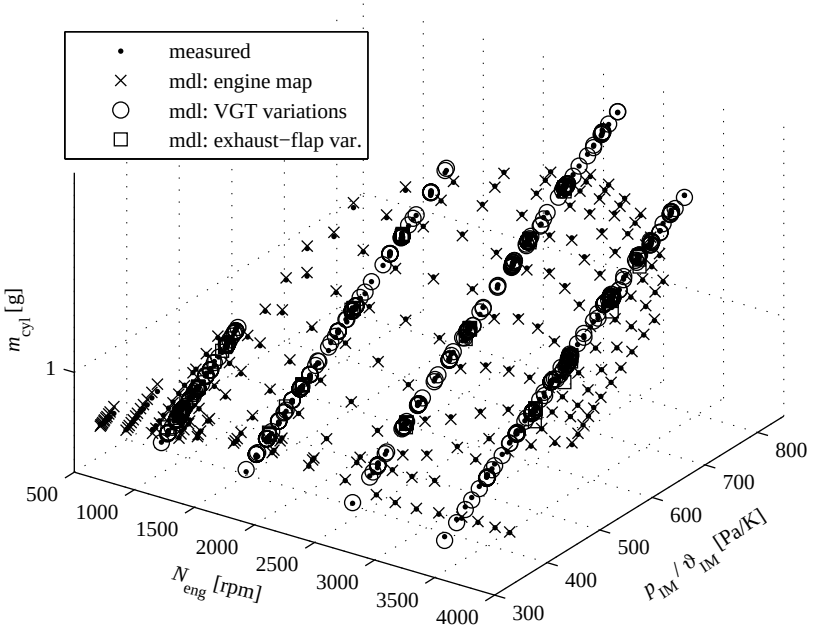

Figure C.5: Volumetric efficiency of engine B. The measured mass entering each cylinder is plotted over the engine speed and the density in the intake manifold. The model is able to accurately reproduce all measurement data. For reasons of confidentiality, the scale for the z-axis is hidden.

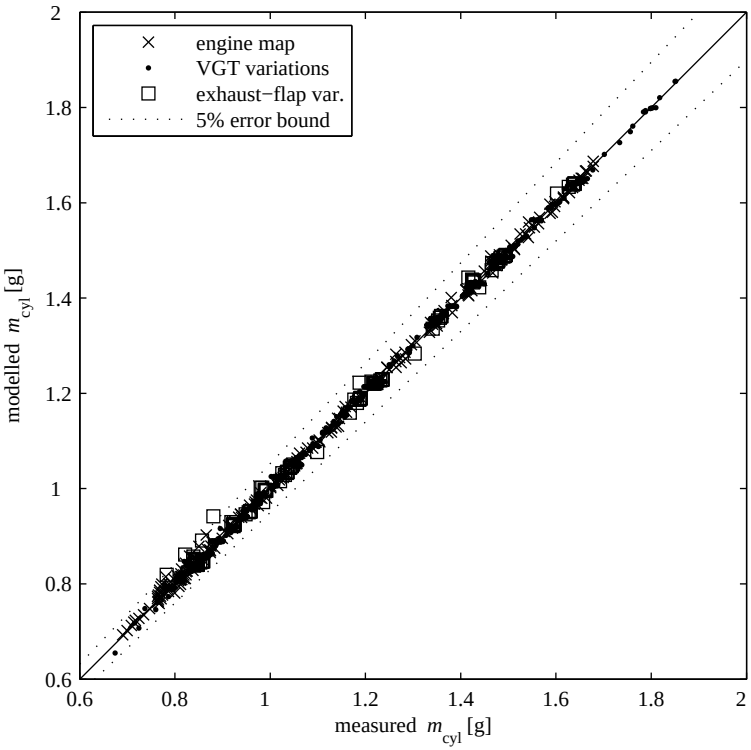

Figure C.6: Volumetric efficiency of engine B. The modelled mass entering each cylinder is plotted against the measurement data.

C.3 Calculating the start of combustion

The start of combustion (SOC) is calculated by means of the heat-release rate obtained by (C.3). From this rate, the cumulative heat release by the combustion is calculated as

$$Q(\varphi) = \int_{\varphi_{\mathrm{SOI}}}^{\varphi} \frac{dQ}{d\varphi'}(\varphi')\, d\varphi'. \tag{C.17}$$

The overall heat (apparently) released by the combustion is denoted by $Q_{\mathrm{comb}} := Q(\varphi_{\mathrm{end}})$, where φ_{end} is chosen as $\min\{\varphi_{\mathrm{EVO}}, \varphi_{\mathrm{SOI}} + \varphi_{\mathrm{comb}}\}$. The parameter φ_{comb} is chosen as 100 degrees crank-angle. Based on the cumulative heat release, the combustion progress is defined as $x_{\mathrm{comb}}(\varphi) := Q(\varphi)/Q_{\mathrm{comb}}$.

The SOC is estimated by the following two steps.

1. Calculate the point $\hat{\varphi}$ at which the combustion progress exceeds a certain threshold $\hat{x}_{\mathrm{comb}}$.

2. Use the slope (i.e. the heat-release rate) at this point to estimate the SOC,

$$\varphi_{\mathrm{SOC}} = \hat{\varphi} - \hat{x}_{\mathrm{comb}} \cdot Q_{\mathrm{comb}} \cdot \left(\frac{dQ}{d\varphi}(\hat{\varphi}) \right)^{-1}. \tag{C.18}$$

A combustion-progress threshold of $\hat{x}_{\mathrm{comb}} = 2\%$ is found to be a reliable choice. The second step compensates for the time required to reach this progress while accounting for the combustion speed. Figure C.7 illustrates this methodology on two cycles (in the same operating point) which substantially differ in the initial combustion speed.

If pilot injections are present, a more sophisticated approach would be necessary to calculate the SOC of the main injection. Especially in the case of a short dwell time between the last pilot and the main injection, the combustion of the pilot injection may extend beyond the SOC of the

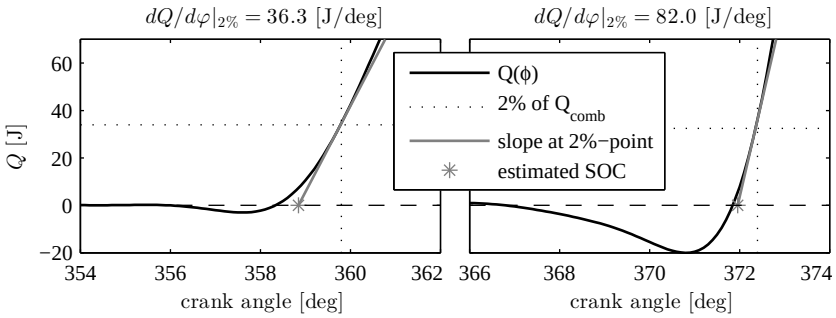

Figure C.7: Methodology of calculating the start of combustion. The operating point at 1900 rpm and 47 mg/cycle is considered for engine B. Although the net heat released by the combustion only differs by 1.25%, the initial combustion is more than twice as fast for the cycle depicted in the right plot. This faster combustion is due to the earlier injection (4° before versus 7.1° after TDC) and a higher rail pressure (1720 versus 940 bar).

main injection. In this case, one possibility is to fit a Wiebe function [80] to the combustion of the pilot injection and subtract this heat-release rate from the calculated one. The start of the main combustion then can be calculated by the procedure just described.

A similar approach can be used to estimate the fraction of the main combustion which burns in the premixed phase. Thereby, a Wiebe function is fitted to the premixed part of the combustion and its area is subtracted from the overall area covered by the heat-release rate. This approach was used to test the dependency of the NO_x emissions on the premixed fraction. Figure C.8 provides three examples, namely a normal combustion and one example for the "NO_x bump" regions (cf. Sec. 2.5) corresponding to an extremely early or late injection.

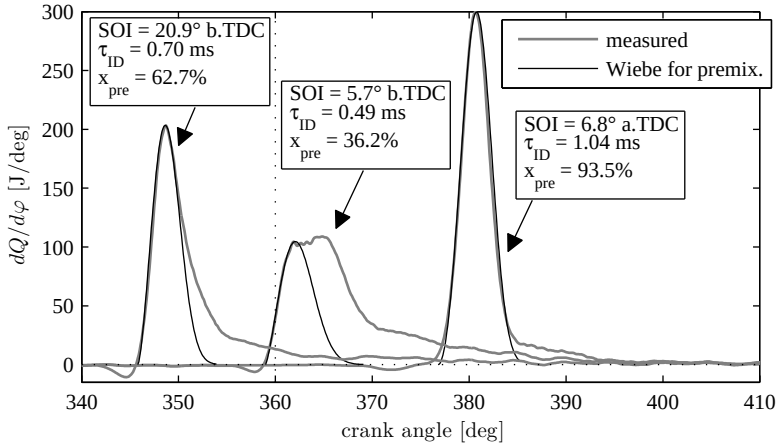

Figure C.8: Application of the Wiebe function to estimate the fraction of the premixed combustion, engine B. The three cycles shown represent a normal and both an extremely early and a late injection timing. The long ignition delays resulting from the latter lead to an almost fully premixed combustion.

D The numerical optimal-control framework

The framework for numerical optimal control is implemented entirely in MATLAB. The readily available interfaces are used to communicate with the external NLP solvers, which are written in Fortran or C/C++. In this section, the parameters and the data that the framework requires as input are explained. Along with this list, the features of the framework are summarised. Before the individual elements are detailed, some general points are highlighted.

- *Why another framework for optimal control?* The existence of various readily available software packages authorises this question. However, the implementation of such a framework from scratch provides several benefits.
  - A suitable degree of flexibility and extendability can be retained in the framework to enable scientific studies. Self-contained packages usually provide limited output and only restricted access to internal quantities. Furthermore, the source code seldom is accessible. Therefore, it is not possible to test custom implementations of specific elements of the framework.

- This accessibility allows for a fast implementation of arbitrary extensions such as different mesh-refinement strategies, time subdivision, regularisation, or MSIM.
- The specific properties of the problem at hand can be taken into account, e.g. time variability or special types of constraints.
- Most software packages do not fully disclose the details of the methods implemented. Similarly, the documentations often lack information on important details. Therefore, it is a tedious, if not impossible, task to perform thorough parametric studies using such packages.
- The knowledge and the understanding gained during the implementation of the framework is immense. Key aspects and critical issues are revealed, and the importance of certain elements manifest themselves.
- These "lessons learned" can be documented and made available to researchers and industrial engineers who just start to plunge into this broad field. This thesis attempts to provide such a transfer of knowledge.
- In accordance with the above points, the framework is implemented in MATLAB. No compilation of code is required, and debugging is fast and simple. This ease of handling facilitates the first implementation and the inclusion or modification of individual components. Furthermore, the framework is quite accessible for new users. Of course, this ease of use cannot offer the same performance as implementations in lower-level, precompiled languages such as Fortran or C/C++ can.

- *Discretisation data.* The collocation nodes, the differentiation matrices and the matrices of the Runge-Kutta coefficients are pre-calculated for Radau (and Lobatto) collocation up to order 500. When solving a single OCP or a series of OCPs, this data has to be loaded only once. Since the functions for the calculation are provided along with the framework, the precalculation of the collocation data to even higher orders is possible.

- *Collocation scheme.* Initially, the framework contained Lobatto and Radau collocation, both in differential as well as in integral formulations. As more and more extensions were added, the efforts were focused on Radau collocation in differential formulation due to the several advantages of this method described in Sec. 3.3. However, the framework itself still is general enough to host any transcription method and, with slight modifications, even multiple shooting approaches.

- *Scaling.* During the development of the framework, several methods were tested to scale the problem. A time-variable scaling according to the initialisation, as well as a normalisation aiming at uniform sensitivities from all inputs to all state variables and throughout the time horizon were tested. However, the best results are achieved by a uniform scaling of the control inputs and the state variables according to their global bounds representing physically plausible ranges. Therefore, this scaling method is used in the current version of the framework.

  If the global lower and upper bounds on variable x are x_{low} and x_{upp}, the normalised variable is $\tilde{x} = (x - x_{\text{low}}) / (x_{\text{upp}} - x_{\text{low}}) \in [0, 1]$. The objective is scaled by the corresponding reference value, and the integral constraints are scaled by their limits. For the path constraints, the user has to specify typical physical ranges. The

framework handles all operations to scale the problem to normalised variables and to re-scale it to physical quantities where required. Therefore, the model functions provided by the user receive and return physical quantities.

D.1 Description of the framework

The purpose of this section is to provide a starting point for fellow researchers and engineers who intend to use, adapt, modify or extend the framework. The folder structure of the framework is outlined in the following list.

\frwkFcns *Core functions of the framework.* The construction of the transcribed OCPs, the initialisation and the solution of the resulting NLPs, as well as the reconstruction and the visualisation of the results are handled by these functions.

\AuxFcns *Auxiliary functions.* Small tasks such as a fast linear and bilinear interpolation of equidistant data, or Lagrange interpolation are performed by the functions in this folder.

\usrFcns *Functions called by the NLP solvers during the solution.* These functions calculate the value and the gradient of the objective function, the constraints and the corresponding Jacobian matrix, as well as the Hessian of the Lagrangian. All information required is obtained from multiple evaluations of the user-supplied model function. Furthermore, the folder contains the functions that calculate the collocation nodes and matrices as well as those to reconstruct a continuous solution from the solution of the NLP.

\SolverOptions This folder contains the function that sets the options of the NLP solvers or generates an external option file if required. The auto-generated files remain in this folder. In the current version of the framework, the options for the solver WORHP have to be set manually in the corresponding XML file.

\OutputFiles Any output of the NLP solvers, if available and specified, is stored in this folder. These files are especially helpful during the

development or the modification of functionalities. For example, user functions can be efficiently debugged with the use of the derivative checkers provided by many NLP solvers.

These folders, as well as the NLP solvers to be used and their licence files, have to be on the MATLAB path. Note that for WORHP, the licence file has to be located in the current working directory of MATLAB. The model functions and all other data required to set up the problem may be located anywhere on the current MATLAB path.

The description of all input parameters below is followed by instructions for initialising and solving the OCP as well as plotting the results.

D.1.1 Problem formulation

The `PROB` struct defines the OCP to be solved. All its fields described here must be set.

PROB.fix_u0 {0, 1} If set to 1, the initial control inputs are fixed at the reference values. If no regularisation or MSIM is applied, this parameter has no influence on a non-subdivided OCP since the left boundary is not a collocation point for Radau collocation.

PROB.fix_x0 {0, 1} If set to 1, the initial state is fixed at the reference.

PROB.QSS_stateVars {*index array*} Indicates which state variables are to be treated as quasi-stationary quantities. Each corresponding differential equation j is replaced by $0 = f_j(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\pi}(t))$.

PROB.dualForm {0, 1} If set to 1, the partially dual formulation introduced in Sec. 3.7.1 is used for all subproblems of the subdivided OCP (or for the non-subdivided original OCP).

PROB.ineqMults {*double array*} Multipliers for the integral constraints when the partially dual formulation is used.

PROB.u_low, PROB.u_upp {*double array* } Global lower and upper bounds on the control inputs.

PROB.x_low, PROB.x_upp {*double array*} Global lower and upper limits on the state variables.

PROB.bandBds.uType {'absolute', 'relative'} Type of the bands around the initial trajectories to which the control inputs are restricted.

PROB.bandBds.u {*double array*} Single-sided magnitude of these band constraints, either as absolute values or as fractions of the initial trajectories.

PROB.bandBds.xType {'absolute', 'relative'} Type of the bands around the initial trajectories to which the state variables are restricted.

PROB.bandBds.x {*double array*} Single-sided magnitude of these band constraints, either as absolute values or as fractions of the initial trajectories.

PROB.locCstrType {*integer array*} Type of the time-variable path constraints, where 0 denotes an equality constraint and 1 an inequality constraint, i.e. $c_i \leq 0$.

PROB.locCstrSc {*double array*} Scaling factors for the path constraints.

PROB.relIneqs {*bool array* } If *true* for any integral inequality constraint, its limit is specified relative to a quantity that is stored as a field in the reference struct.

PROB.relFields {*string array*} The names of the fields of the reference struct that represent the quantities to which each relative inequality constraint is related.

PROB.IneqLims {*double array*} Limits for the integral constraints. If `NaN` is specified, the value of the reference is used.

PROB.mdlFcn {*string*} The name of the model function. The function is a MATLAB function that is within the current path. Its signature is

```
f_ext = mdlFcn(t, x, u, pi, params)
```

The arguments are the time, the state variables, the control inputs, the time-variable parameters, and the struct containing the model parameters. The output of the model function is a vector that contains the RHS of the model ODEs $\boldsymbol{f}$, the integrands of the integral constraints $\boldsymbol{g}$ and the objective L, as well as the nonlinear path constraints $\boldsymbol{c}$, i.e.

$$\boldsymbol{f}_{\text{ext}} = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{g} \\ L \\ \boldsymbol{c} \end{pmatrix}. \tag{D.1}$$

If the framework is used with a solver that needs to be able to evaluate only the objective and its gradient, the function

```
L = mdlFcn_objOnly(t, x, u, pi, params)
```

has to be available, too. All solvers currently supported, except for SNOPT, require the availability of this objective-only model function.

PROB.mdlParams {*struct*} A struct that contains all model parameters. This struct is passed to the model functions as the last argument.

PROB.mdlValMaps {*string*} Full path to a .mat file that contains lookup maps constraining the range on which the model is trusted. The admissible control and state spaces are restricted to this region. The file has to provide the vectors `x_vec` and `y_vec` which span the maps and the four three-dimensional arrays containing the map data. The latter have to be named `u_min_MAPS`, `u_max_MAPS`, `x_min_MAPS`, and `x_max_MAPS`.

PROB.mdlValX, PROB.mdlValY {*double array*} Actual values of the abscissa and the ordinate for the validity-map lookup at the time instances in `PROB.t_ref`.

PROB.mdlVal_ctrl, PROB.mdlVal_state {*index array*} These arrays define to which control inputs and state variables the model-validity limits are actually applied.

PROB.inertiaMult {*double*} Scaling factor for the dynamic parameters of the model. A value of < 1 speeds up the model dynamics.

PROB.inertiaFields {*string array*} Names of the fields in the parameter struct `PROB.mdlParams` that are scaled by `PROB.inertiaMult`.

PROB.t_ref {*double array*} Time vector of the reference / initialisation.

PROB.u_ref, PROB.x_ref {*double array*} Reference / initial trajectories of the control inputs and the state variables.

PROB.objIneq_ref {*double array*} Reference / initial cumulative trajectories of the integral constraints and the objective.

PROB.p_ref {*double array*} Time-variable parameters.

PROB.addRefs {*string array*} Names of fields in the `PROB` struct that are copied to the OCP struct as additional reference trajectories. Only these fields can be referenced, e.g. by `PROB.relFields`.

PROB.p_pairs {*integer matrix*} A matrix with three columns. Each row $[i, j, \text{TYPE}]$ defines a pair of time-variable parameters that share an infinitesimal relation, namely

$$\pi_i = \begin{cases} \dot{\pi}_j, & \text{if TYPE } = 1, \\ \int \pi_j \, dt, & \text{if TYPE } = 2. \end{cases} \tag{D.2}$$

Only the time-variable parameter π_j has to be specified, while the entries for π_i can be filled with zeros in `PROB.p_ref`. The actual discretisation scheme used by the transcription is applied to calculate π_i from π_j to ensure a consistent discretisation of the continous-time problem. In the case of an integral relation, the initial value $\pi_i(0)$ specified by the user is enforced to locate the indefinite integral.

D.1.2 Setup of the algorithms

The `ALGO` struct defines all the algorithmic options such as the transcription method, the derivative calculation, or the NLP solver to be used. All its fields described here must be set.

ALGO.parallGrade {0, 1, 2, 3} Parallelisation of the model evaluations, see Sec. 3.4.4. Valid choices are no parallelisation (0), discretisation-point scheduled parallelisation (1), partition of the discretisation points into equally sized sets (2), and full parallelisation including the construction of the Jacobian (3). Currently, regularisation and MSIM are not implemented for option (3).

ALGO.discr_method {'diff_coll'} Integral collocation ('int_coll') currently not available.

ALGO.discr_type {'RadIIA'} Lobatto ('LobIIIA') currently not available.

ALGO.uContMode {'u-lin', 'u-cont'} Method to construct the continuous control signals. If the option 'u-lin' is set, a piecewise-linear interpolation between the collocation nodes is applied. In the case of 'u-cont', a polynomial is fitted to the values of the control inputs at the collocation points. Since the control inputs at the left boundary does not influence the discretised ODEs, the control polynomials have one order less than the state polynomials).

ALGO.derivsMode {'FD', 'CD', 'AD', 'usrJac-FD', 'usrJac-CD', 'usrHess'} Method to calculate the model Jacobian and Hessian. Forward ('FD') and central finite differences ('CD') as well as algorithmic differentiation ('AD') are provided. For the latter, the model function cannot use matrix operations, and vectors have to be constructed by concatenation instead of indexing. If the model function provides the (exact) Jacobian as a second return value, 'usrJac-FD' and 'usrJac-CD' calculate the Hessian by applying forward or centered FD to the gradient of each element. The last option 'usrHess' expects the model function to also provide the exact Hessian as third return value. The last three options are not validated as yet.

ALGO.derivsPert {*double*} Perturbation for FD.

ALGO.init_tRes {*double*} Initial uniform lengths of the integration intervals.

ALGO.init_tRes_OL {*double*} Ditto for the overlap phases (if time subdivision with overlap is used).

ALGO.init_order {*integer*} Initial uniform collocation order on all integration intervals.

ALGO.c_reg {*double*} Regularisation coefficient c_{reg}, see Sec. 3.8.

ALGO.MSIM_mode {0,1,2} Type of the multiple-shooting imitation mode described in Sec. 3.9. Off (0), piecewise constant control (1), piecewise linear control (2).

ALGO.MSIM_uInds {*index array*} Indices of the control inputs to which MSIM is applied.

ALGO.usrJac {0,1} Provide the Jacobian to the NLP solver.

ALGO.usrHess {0,1} Provide the Hessian of the Lagrangian to the NLP solver (and thus use the exact Newton method, if applicable).

ALGO.HessUpd {'BFGS', 'SR1', {'L-BFGS', *X*}} Type of the quasi-Newton update used if `ALGO.usrHess` = 0. For the limited-memory BFGS update, *X* is a positive integer which specifies the number of the last steps used to approximate the Hessian.

ALGO.preFeas {0,1,2} "Pre-feasibilisation" mode. Specified control inputs can be fixed at their initial trajectories. Mainly used for debugging if the problem is infeasible. If all control inputs are fixed, a "simulation" on the specified grid is executed. Off (0), pre-feasibilisation only (1), perform pre-feasibilisation and run optimisation starting with the (hopefully) feasible initialisation obtained by the pre-feasibilisation.

ALGO.preFeasFixU {*index array*} Specifies which control inputs are fixed during pre-feasibilisation mode.

ALGO.preFeasIneqs {*double array*} Factors to relax the limits for the integral inequality constraints for pre-feasibilisation. If all control inputs are fixed, the optimisation has no degrees of freedom left to satisfy additional constraints, and thus they should be eliminated by using a large value for these factors.

ALGO.cautMode {*double*} For debugging. Limits all NLP variables to a relative range of $1\pm$ `ALGO.cautMode` around their initial values. Off if set to -1.

ALGO.NLP_solver {'SNOPT', 'IPOPT', 'WORHP', 'KNITRO-AS', 'KNITRO-IPDIR', 'KNITRO-IPCG'} The solver selected as well as a valid licence file have to be on the MATLAB path.

ALGO.NLP_iter {*integer*} Maximum number of NLP iterations.

ALGO.NLP_tol {*double*} Tolerance to which the NLP is solved.

ALGO.NLP_scale {0,1} Switches the internal scaling of the problem by the NLP solver on or off.

All specific options for the NLP solvers are set directly in the files in the *SolverOptions* folder.

D.1.3 Time subdivision

Only few options are available for the time subdivision. In fact, each iteration of the high-level optimisation is started manually (or by ad-hoc loops). Therefore, the options described here may be changed between any two consecutive iterations, see Sec. D.1.4.

SUBD.tSplit {*double array*} Points in time where the time horizon is subdivided. If set to an empty array [ ], no subdivision is applied.

SUBD.lambdaUpd {'sens', 'approx', 'accel'} Mode of the multiplier update. The exact update invoked by 'sens' relies on sensitivites calculated by FD on the NLP, 'approx' calls the approximate update, and 'accel' invokes the acceleration of the approximate update. The acceleration is applicable only after at least $n_g + 1$ iterations of the approximate update were performed.

SUBD.lambdaUpdPars {*various*} Array of parameter values, specific to each multiplier-update mode.

- *sens*: {MAXSENSR, RELCHLIM, PERT}. The sensitivity ratio between the sensitivities on the intervals with and without overlap is limited to MAXSENSR in both directions. A maximum relative change of the limits for the integral inequality constraints is imposed by RELCHLIM. Finally, PERT specifies the relative perturbation of these limits to calculate the sensitivities.
- *approx*: {POW, XIR, RELCHLIM}. Power of the proportionality between multipliers and integral inequality constraints, reduction of the limits for inactive constraints, limit for the relative change (as for *sens*).
- *accel:* {NITER, EXPDEC}. Number of most recent iterations used to construct the linear models, and the factor for the exponential decay to weight the regression.

SUBD.elastInitFacs {*double array*} Initially relaxes the integral inequality constraints on each subproblem by these factors. This relaxation may be necessary if the problem is very "local" and a good initialisation is not available.

D.1.4 Solution process

Once the three structs `PROB`, `ALGO` and `SUBD` are defined, the problem is initialised and executed by procedural code and some function calls. All relevant information is gathered, and additional data required for the solution of the OCP is constructed. The result is a new struct `SDOCP` that contains all the information on the problem, the algorithms and the time subdivision. The following list outlines all functions concerning the initialisation, the first solution run and the reconstruction, as well

as further iterations on the OCP such as a mesh refinement or high-level iterations if time subdivision is applied.

- `SDOCP = getPathBds(PROB, SDOCP, PLOT).`
  Defines the simple bounds on the control inputs and the state variables. The global limits, the model-validity region, and the bounds around the initial values are considered. At each discretisation point, the most restricting bound is imposed. A plot for the user to check the resulting bounds may be requested by setting `PLOT` to 1.

- `SDOCP = initSDOCP(SDOCP, PROB, SUBD.elastInitFacs).`
  Initialises the subproblems. The collocation grid is constructed, the transcription method is initialised, the scaling of the problem is performed (by the global bounds), and the NLP variables are initialised by the reference trajectories.

- ```
  SDOCP = runIterSDOCP(SDOCP, ALGO.preFeas, ...
                ALGO.preFeasFixU, ALGO.preFeasIneqs, ...
                ALGO.cautMode, QUIT_ON_FAIL).
  ```
  Solves the subproblems sequentially. The last argument is a boolean variable. If true, the solution process is terminated whenever the solution of a subproblem fails. The solutions of the subproblems solved previously are returned in the `SDOCP` struct, and thus the solution process may be restarted at this point.

- `SDOCP = mergeSDOCP(SDOCP).`
  Merges the solutions of the subproblems to a global solution.

- ```
  plotSDOCPresults(SDOCP, TIMESHIFT, PLOTMODE, ...
                PLOTINIT, PLOTWHAT, DRAGEVENTS, LINECOL).
  ```
  Plots the results of the most recent solution run. If `TIMESHIFT`

is true, the time of the time horizon is shifted such that it starts at $t = 0$. PLOTMODE is a string consisting of two parts connected by a hyphen, {'glob', 'indiv'} - {'cont', 'both'}. The first part indicates whether the global solution is plotted or the solutions of the individual subproblems are displayed. The second part defines whether only the reconstructed continous solution or also the discrete solution is shown. For the option 'indiv-cont', the string '-fade' may be appended to plot the overlap phases in a lighter color.

If PLOTINIT is true, the initialisation of this solution run is also plotted. PLOTWHAT is a struct that defines which signals are plotted. It contains the fields 'lC','u', and 'x', which are index arrays. The field 'objIneq' is a boolean variable that decides whether the cumulative objective and the integral inequality constraints are plotted. DRAGEVENTS is an optional argument. It contains the start and end times of the drag phases in a two-column matrix. These phases will be indicated in all plots. Finally, LINECOL is a string that specifies the line color of the plots.

- `SDOCP = L_update(SDOCP, SUBD.lambdaUpd, ... SUBD.lambdaUpdPars).`
  Updates the limits and the unified multipliers for the integral inequality constraints of a subdivided problem. If there is only one subproblem, i.e. the time horizon is not divided, provide 'none' and an empty array [ ] as second and third arguments.

- `SDOCP = refineMesh(SDOCP, REFMODE, REFPARS, REFSPS).`
  Mesh refinement. The type of the refinement REFMODE and the corresponding parameters REFPARS are described in the following list.

  – 'none', [ ]. No mesh refinement.

  - 'h-unif', NREF. All collocation intervals are split into NREF intervals, both using the same collocation order as the original interval.
  - 'p-unif', NREF. The collocation order is increased by NREF on all intervals.
  - 'locTruncErr', [RELTOL, SAFEK, MINSTEP, PLOTMODE, USEKIN]. The refinement described in Sec. 3.5. RELTOL is the desired relative tolerance, SAFEK is the "saftey margin" k, and MINSTEP is the minimum length of any collocation interval. PLOTMODE is either -1 (no plots at all), 0 (waitbar during the refinement), 1 (real-time plot of the refinement process), or 2 (final plot only). If the last parameter USEKIN is true, KINSOL is used instead of MATLAB's fsolve. The SUDIALS package has to be installed in order to use this option.

  The index array REFSPS designates for which subproblems the mesh refinement is executed.

- SDOCP = initIterSDOCP(SDOCP, PROB, SENSINIT, INITMODE). Initialisation of the next high-level iteration. If SENSINIT is true, the sensitivities calculated during the previous iteration are used for the initialisation of the subproblems. This option is only applicable if the multiplier-update 'sens' is chosen and is of an experimental nature in the current version of the framework.

  The type of the initialisation specified by INITMODE is either 'ref', 'prevDir' or 'prevCont'. The first initialises the subproblems by the reference trajectories, whereas the latter two use the solution of the previous iteration. Either a linear interpolation of the discrete solution or the actual reconstructed continuous solution is used.

After applying a multiplier update and/or a mesh refinement and initialising the next iteration, another solution run is executed by calling `runIterSDOCP` and `mergeSDOCP` again. Plots concerning the high-level optimisation of a subdivided problem are generated by

```
plotSDOCP_highlevel(SDOCP, ITERNOS, LINESPECS, ...
                    TIMESHIFT)
```

`ITERNOS` is an index array that defines which high-level iterations are plotted. The line styles corresponding to these iterations is defined in the string array `LINESPECS`.

D.2 Open points and future work

The framework is still under development and thus is far from providing a fully reliable "fire-and-forget" solution of arbitrary OCPs. However, as mentioned above, the goal of developing another transcription framework was to gain insights into the most critical aspects of the direct-transcription approach to solve OCPs numerically. According to J.T. Betts [38], "[T]he most successful direct transcription codes, whether academic or industrial, have evolved over time and have required the efforts of a number of people." Therefore, the most important points that require refinement, as well as directions for future extensions are outlined in the following list.

- A single, clean initialisation function should be written that copies all data from the three set-up structs to the OCP struct and defines all global variables. No more procedural code should be required which has to be copied manually to each file invoking the optimisation.

- Implement the exploitation of the model sparsity in the first and second-order derivatives. An automatic detection of the sparsity patterns of the model functions and of the index sets would be required.

- Include an option that ensures that the bounds on the control inputs and the state variables are honoured during the calculation of the model derivatives. For physically constrained quantities such as the fuel injection, which cannot be negative, only a variation in one direction is plausible at the limit. In such cases, a single-directed scheme has to be applied, also for the second derivatives.

- The "dense indexing" during the construction of the Jacobian and the Hessian of the NLP in sparse matrix format should be replaced by a thorough construction of these sparse matrices. A substantial speedup could result for large problems for which the model evaluations are fast.

- Instead of saving the reconstructed continous solution on a fine time grid, the barycentric weights of the Lagrange interpolation of the collocation nodes should be saved to the `SDOCP` struct. These coefficients can be used to evaluate the continous solution at the points required, e.g. to initialise the refined problem or when evaluating and plotting the final result. Especially when very long time horizons are considered and multiple high-level iterations are executed, the barycentric weights require considerably less memory than the full trajectories.

- More sophisticated refinement methods should be implemented, e.g. an adaptive *hp* method. For the *h*-refinement implemented in the current version, the constraint violation should be considered during the refinement.

- A more thorough analysis of singular arcs and regularisation should be performed. Furthermore, an automatic selection of the regularisation parameter could be implemented.

- The generality of the framework w.r.t. the problem formulation could be increased further. Other types of constraints could be implemented such as cyclic constraints or a free end time. Although such extensions are simple to implement in the framework by a person that understands its structure, a consistent propagation of the changes through all the framework functions is a tedious and error-prone task.

- Making the code publicly available should be a primary goal of any further effort. However, before the framework can be used by the public, at least three points need to be improved upon.

    1. The generality of the framework w.r.t. the transcription method should be omitted, i.e. Radau collocation should be used as the unique method. This restriction to one specific method enables a vast reduction of the structural complexity of several functions. Furthermore, all functionalities concerning the time subdivision could be omitted, since this feature is applicable only to a narrow class of OCPs. Again, the complexity of the framework could be substantially reduced by this step.

    2. Recurring pieces of code should be consolidated in specific functions. For example, the various parallelisation schemes for the model evaluations, or the construction of the Jacobian are hard-coded in each user-function type in the current version of the framework. Therefore, when an extension is added, the changes in the user function have to be repeated several times in order to port the new functionality to all NLP solvers supported. This procedure is time consuming and prone to errors. The core functionalities should thus be handled by specific functions that are used to modularly construct any specific user function required by a given NLP solver.

    3. The minimalist documentation provided in this Appendix has to be extended. Furthermore, at least two or three examples should be provided with the framework code to provide an illustrative starting point for future users.

Besides the natural, "evolutionary" extensions and modifications outlined in the list above, two "revolutionary" steps could be aspired.

- Development of a custom NLP solver. The solver should exploit the structure of the NLPs resulting from the direct transcription of continous OCPs by Radau collocation. A first step in this direction could be to extend an existing open-source code by the partitioned quasi-Newton update of the Hessian of the Lagrangian described in Sec. 3.4.3.

- Implementation of the framework and the model functions in a precompiled language such as Fortran or C/C++. The evaluations of the model functions of the problem considered in this thesis are found to be at least three times faster in C than in MATLAB [S.5]. Furthermore, some efficient implementations of algorithmic differentiation do exist for C. An alternative could be an automated code generation based on the MATLAB implementation. Finally, "julia" (http://julialang.org) is a promising new high-level language similar to MATLAB, which uses a just-in-time compiler to enable a performance similar or even better than the aforementioned low-level languages. However, the state-of-the-art NLP solvers do not provide any interfaces to this new language as yet.

E Student projects supervised

[S.1] MATLAB**/Simulink Simulation Performance**
Leisibach Ronald, spring term 2010, semester project

The various ways of implementing models in Simulink are compared w.r.t. execution speed. Moreover, the impact of the accelerator modes is analysed. As long as appropriate solvers and parameter settings are used, all approaches are found to exhibit similar performance.

[S.2] **Globally Optimal Control of a Diesel Engine**
Volken Florian, spring and fall terms 2010, master thesis

Iterative dynamic programming is identified as the most promising heuristic approach to solve the optimal control problem for diesel engines. Furthermore, static optimisation of the control maps is considered. The results indicate that heuristic methods for optimal control struggle with the dimensions of the problem and often produce nonsmooth solutions.

[S.3] **Behavior of Diesel Engines During Transient Driving-Cycles**
Wyder Thierry, spring term 2011, bachelor thesis

The dynamics of a diesel engine operated on a transient driving cycle is analysed. A clustering approach is used to categorise the transient events occurring throughout the cycle. However, due to the variability of heavy-duty transient cycles and the vast amount of possible categorisation criteria, an unambiguous categorisation is difficult to achieve .

[S.4] **Air-Path Strategies for a Diesel Engine**
Nüesch Sandro, fall term 2011, semester project

A control-parametrisation approach is used to find the optimal transient control strategy for the air path of a diesel engine. The NO_x-soot tradeoff over a short transient event is derived and analysed. The difficulties encountered during the work highlight the important roles of the model and the transcription method in the context of numerical optimal control.

[S.5] **Trajectory Optimisation for a Diesel Engine: Pseudospectral Methods**
Liniger Alexander, fall term 2011, semester project

The open-source MATLAB toolbox GPOPS-I is used to assess the applicability of pseudospectral methods to solve the optimal control problem for diesel engines. Furthermore, PSOPT is used to compare a similar implementation in C++. Pseudospectral methods are found to be a promising approach w.r.t. the accuracy of the solution as well as the computational performance. The engine model implemented in C++ runs approximately three times faster than its implementation in MATLAB.

[S.6] **Static Optimisation of Diesel Engine Maps**
Bapst Roman and Jakob Marcel, spring term 2012, bachelor thesis

Comparison of linear, diagonal-quadratic and full-quadratic models for the fuel consumption and the emissions, when applied to static optimisation. Diagonal-quadratic models are found to be almost as accurate as

the full quadratic ones. Furthermore, for these models it is straightforward to impose positive definiteness. Consequently, a reliable optimisation procedure can be derived. In addition, a weighting matrix is introduced to include the information on the transient driving cycle in the static optimisation.

[S.7] **Trajectory Optimisation for a Diesel Engine: Local Transcription Methods**
Li Bing, spring term 2012, semester project

Compared to pseudospectral methods, local direct transcription applies relatively low-order integration schemes to small integration intervals. Custom-coded Jacobian matrices are implemented for the trapezoidal discretisation scheme, and first results are presented.

[S.8] **Trajectory Optimisation for a Diesel Engine: Direct Multiple Shooting**
Skoda Marcel, spring term 2012, semester project

The SUNDIALS suite of ODE solvers is used to implement a direct multiple-shooting approach for optimal control. Utilisation of the integrated sensitivity calculation is crucial to obtain consistent derivatives. Concerning the computational performance, multiple shooting is found to be substantially inferior to direct transcription approaches.

[S.9] **Trajectory Optimisation for a Diesel Engine: Causal Optimal Control**
Gehlen Manuel, spring term 2012, semester project

Receding-horizon optimal control is applied to assess the influence of relying on causal information only. The emissions have to be included in the objective, weighted by their corresponding Lagrange multipliers. Various prediction schemes for the operating point are tested (constant, linear,

Markov chain), and a linear prediction of the operating point over a fraction of the prediction horizon is found to be a good choice. The methodology presented can be used to derive causal benchmark solutions to assess the performance of implementable control structures.

[S.10] **Model-Predictive Control of Diesel Engines**
Graf Plessen Mogens, fall term 2012, semester project

First steps are taken towards a fast implementation of the receding-horizon optimisation, relying on a prediction of the operating-point profile. Above all, the choice of the discretisation grid is analysed in detail. Furthermore, a combination of cascaded control and receding-horizon optimisation is tested.

[S.11] **Optimisation-Oriented Modelling of an SCR Catalyst and Optimal Control of a Complete Diesel-Engine System**
Skoda Marcel, fall term 2012 and spring term 2013, master thesis

An optimisation-oriented model has to capture all relevant trends, be quantitatively accurate, and exhibit a simple structure to allow a fast evaluation. A first attempt at constructing such a model for the SCR system is presented, highlighting the critical points. The modelling itself does not pose a major challenge, but a fast and reliable identification of the model parameters is a demanding task and requires all insights into the underlying physics to be exploited.

[S.12] **Preliminary Studies on MPC for Diesel Engines**
Bapst Roman and Jakob Marcel, fall term 2013, semester project

A simple framework for nonlinear model-predictive control (MPC) is implemented based on direct transcription. Within that approach, the main computational cost is the calculation of the Jacobian matrices of the model functions. By using only a single model linearisation at the current ini-

tial state, the execution speed is significantly reduced, while the reference-tracking performance of the MPC is not substantially affected. Another approach to speed up the execution of the MPC is to pre-calculate the model Jacobians on a multidimensional grid. A principal-component analysis is applied to reduce the dimensionality of this grid, and a study on the necessary resolution is executed. The performance of the two MPC versions are compared to a PI controller that is tuned according to the same objective function used to formulate the MPC problem.

Bibliography

[1] *Community portal for automatic differentiation*, http://www.autodiff.org (last visited April 2013).

[2] *The HSL mathematical software library*, http://www.hsl.rl.ac.uk (last visited August 2013).

[3] *METIS – serial graph partitioning and fill-reducing matrix ordering*, http://glaros.dtc.umn.edu/gkhome/metis/metis/overview (last visited August 2013).

[4] *MUMPS homepage*, graal.ens-lyon.fr/MUMPS (last visited April 2013).

[5] *PARDISO homepage*, http://www.pardiso-project.org (last visited August 2013).

[6] *SUNDIALS homepage*, https://www.llnl.gov/casc/sundials (last visited February 2013).

[7] S. M. Aithal, *Modeling of NOx formation in diesel engines using finite-rate chemical kinetics*, Applied Energy **87** (2010), 2256 – 2265.

[8] S.M. Aithal and D. Upadhyay, *Feasibility study of the potential use of chemistry based emission predictions for real-time control of modern diesel engines*, Applied Energy **91** (2012), 475 – 482.

[9] Mohamed F. Al-Dawody and S.K. Bhatti, *Optimization strategies to reduce the biodiesel NOx effect in diesel engine with experimental verification*, Energy Conversion and Management **68** (2013), 96 – 104.

[10] Ezio Alfieri, *Emissions-controlled diesel engine*, Ph.D. thesis, ETH Zurich, 2009.

[11] R. Amacher, J. Asprion, G. Ochsner, H. Tevaearai, M. J. Wilhelm, A. Plass, A. Amstutz, S. Vandenberghe, and M. Schmid Daners, *Numerical optimal control of turbodynamic ventricular assist devices*, Bioengineering **1** (2014), 22 – 46.

[12] Magnus Andersson, Bengt Johansson, Anders Hultqvist, and Christof Noehre, *A predictive real time NOx model for conventional and partially premixed diesel combustion*, SAE technical paper **2006-01-3329** (2006).

[13] C. Arcoumanis, *Measurement and prediction of transient NOx emissions in DI diesel engines*, Combustion in Engines, vol. 1992-10, Proceedings of the Institution of Mechanical Engineers, no. C448/039, 1992, pp. 97 – 105.

[14] Jean Arrègle, J. Javier López, Carlos Guardiola, and Christelle Monin, *Sensitivity study of a NOx estimation model for on-board applications*, SAE technical paper **2008-01-0640** (2008).

[15] ———, *On board NOx prediction in diesel engines: A physical approach*, Automotive Model Predictive Control, Springer Berlin Heidelberg, 2010, pp. 26 – 36.

[16] Uri M. Ascher and Linda R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*, SIAM, Philadelphia, 1998.

[17] J. Asprion, G. Mancini, S. Zentner, Ch. H. Onder, N. Cavina, and L. Guzzella, *A framework for the iterative dynamic optimisation of diesel engines: numerical methods, experimental setup, and first results*, WIT Transactions on Ecology and the Environment (C.A. Brebbia, E.R. Magaril, and M.Y. Khodorovsky, eds.), vol. 190, WIT Press, Southampton, 2014.

[18] Jonas Asprion, Oscar Chinellato, and Lino Guzzella, *A fast and accurate physics-based model for the NO_x emissions of Diesel engines*, Applied Energy **103** (2013), 221 – 233.

[19] _____, *Optimisation-oriented modelling of the NO_x emissions of a Diesel engine*, Energy Conversion and Management **75** (2013), 61 – 73.

[20] _____, *Optimal control of diesel engines: Numerical methods, applications, and experimental validation*, Accepted for publication in Mathematical Problems in Engineering (2014), Special Issue on Advanced Control and Optimization with Applications to Complex Automotive Systems.

[21] Jonas Asprion, Oscar Chinellato, Christopher H. Onder, and Lino Guzzella, *From static to dynamic optimisation of diesel-engine control*, 52nd IEEE Conference on Decision and Control, Florence, 2013, pp. 6824 – 6829.

[22] Jonas Asprion, Christopher H. Onder, and Lino Guzzella, *Including drag phases in numerical optimal control of Diesel engines*,

7th IFAC Symposium on Advances in Automotive Control, Tokyo, 2013, pp. 489 – 494.

[23] D.N. Assanis, Z.S. Filipi, S.B. Fiveland, and M. Syrimis, *A predictive ignition delay correlation under steady-state and transient operation of a direct injection diesel engine*, Journal of Engineering for Gas Turbines and Power **125** (2003), 450 – 457.

[24] Chris Atkinson, Marc Allain, and Houshun Zhang, *Using model-based rapid transient calibration to reduce fuel consumption and emissions in diesel engines*, SAE technical paper **2008-01-1365** (2008).

[25] Mohamed Ayeb, Heinz J. Theuerkauf, and Thomas Winsel, *SI engine emissions model based on dynamic neural networks and D-optimality*, SAE technical paper **2005-01-0019** (2005).

[26] G. Bashein and M. Enns, *Computation of optimal controls by a method combining quasi-linearization and quadratic programming*, International Journal of Control **16** (1972), 177 – 187.

[27] Z. Bazari, *A DI diesel combustion and emission predictive capability for use in cycle simulation*, SAE technical paper **920462** (1992).

[28] _____, *Diesel exhaust emissions prediction under transient operating conditions*, SAE technical paper **940666** (1994).

[29] V.M. Becerra, *PSOPT optimal control solver user manual, release 3*, 2010.

[30] _____, *Solving complex optimal control problems at no cost with PSOPT*, Proc. IEEE Multi-conference on Systems and Control, Yokohama, 2010, pp. 1391 – 1396.

[31] Jesús Benajes, Santiago Molina, and José M. García, *Influence of pre- and post-injection on the performance and pollutant emissions in a HD diesel engine*, SAE technical paper **2001-01-0526** (2001).

[32] David A. Benson, Geoffrey T. Huntington, Tom P. Thorvaldsen, and Anil V. Rao, *Direct trajectory optimization and costate estimation via an orthogonal collocation method*, Journal of Guidance, Control, and Dynamics **29** (2006), 1435 – 1440.

[33] Michael Benz, Markus Hehn, Christopher H. Onder, and Lino Guzzella, *Model-based actuator trajectories optimization for a diesel engine using a direct method*, Journal of Engineering for Gas Turbines and Power **133** (2010), no. 3, 032806–1 – 032806–11.

[34] Michael Benz, Christopher H. Onder, and Lino Guzzella, *Engine emission modeling using a mixed physics and regression approach*, Journal of Engineering for Gas Turbines and Power **132** (2010), no. 4, 042803–1 – 042803–11.

[35] Jean-Paul Berrut and Lloyd N. Trefethen, *Barycentric Lagrange interpolation*, SIAM Review **46** (2004), 501 – 517.

[36] John T. Betts, *Survey of numerical methods for trajectory optimization*, Journal of Guidance, Control, and Dynamics **21** (1998), 193 – 207.

[37] _____, *Practical methods for optimal control and estimation using nonlinear programming*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, 2010.

[38] J.T. Betts, S.L. Campbell, and A. Engelsone, *Direct transcription solution of optimal control problems with higher order state*

constraints: theory vs practice, Optimization and Engineering **8** (2007), no. 1, 1–19.

[39] Lorenz T. Biegler, *Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation*, Computers and Chemical Engineering **8** (1984), 243 – 248.

[40] ———, *An overview of simultaneous strategies for dynamic optimization*, Chemical Engineering and Processing: Process Intensification **46** (2007), no. 11, 1043 – 1053.

[41] Christian H. Bischof, H. Martin Bücker, André Vehreschild, and Johannes Willkomm, *Automatic differentiation for Matlab (ADiMat)*, MATLAB-Day, Aachen, 2012.

[42] H.G. Bock and J. Plitt, *A multiple shooting algorithm for direct solution of optimal control problems*, Proceedings of the 9th IFAC World Congress, Budapest, 1984, pp. 243 – 247.

[43] Craig T. Bowman, *Kinetics of pollutant formation and destruction in combustion*, Progress in Energy and Combustion Science **1** (1975), 33 – 45.

[44] Indranil Brahma, Mike C. Sharp, and Tim R. Frazier, *Empirical modeling of transient emissions and transient response for transient optimization*, SAE technical paper **2009-01-1508** (2009).

[45] Daniel Brand, *Control-oriented modeling of NO emissions of SI engines*, Ph.D. thesis, ETH Zurich, 2005.

[46] A. Burcat, *Prof. Burcat's thermodynamic data*, 2011, http://garfield.chem.elte.hu/Burcat/burcat.html (last visited September 2013).

[47] Christof Büskens and Helmut Maurer, *SQP-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control*, Journal of Computational and Applied Mathematics **120** (2000), no. 1–2, 85 – 108.

[48] Christof Büskens and Denis Wassel, *The ESA NLP solver WORHP*, Modeling and Optimization in Space Engineering (Giorgio Fasano and János D. Pintér, eds.), Optimization and Its Applications, vol. 73, Springer New York, 2013, pp. 85 – 110.

[49] J.C. Butcher, *Numerical methods for ordinary differential equations*, John Wiley & Sons, Winchester, 2003.

[50] Richard H. Byrd, Jorge Nocedal, and Richard A. Waltz, *KNITRO: An integrated package for nonlinear optimization*, Large-Scale Nonlinear Optimization (G. Pillo and M. Roma, eds.), Nonconvex Optimization and Its Applications, vol. 83, Springer US, 2006, pp. 35 – 59.

[51] Claudio Canuto, M. Youssuff Hussaini, Alfio Quarteroni, and Thomas A. Zang, *Spectral methods: Fundamentals in single domains*, Springer Berlin, 2006.

[52] John F. Cassidy, *A computerized on-line approach to calculating optimum engine calibrations*, SAE technical paper **770078** (1977).

[53] Shi-Chung Chang, Tsu-Shuan Chang, and Peter B. Luh, *A hierarchical decomposition for large-scale optimal control problems with parallel processing structure*, Automatica **25** (1989), no. 1, 77 – 86.

[54] Bruce A. Conway, *A survey of methods available for the numerical optimization of continuous dynamic systems*, Journal of Optimization Theory and Applications **152** (2012), 271 – 306.

[55] Christopher L. Darby, William W. Hager, and Anil V. Rao, *Direct trajectory optimization using a variable low-order adaptive pseudospectral method*, Journal of Spacecraft and Rockets **48** (2011), no. 3, 433 – 445.

[56] Christopher L. Darby, William W. Hager, and Anil V. Rao, *An hp-adaptive pseudospectral method for solving optimal control problems*, Optimal Control Applications and Methods **32** (2011), no. 4, 476 – 502.

[57] Arturo de Risi, Paolo Carlucci, Teresa Donateo, and Antonio Ficarella, *A combined optimization method for common rail diesel engines*, American Society of Mechanical Engineers, Internal Combustion Engine Division (Publication) ICE, vol. 38, 2002, pp. 243 – 250.

[58] Biplab K. Debnath, Niranjan Sahoo, and Ujjwal K. Saha, *Adjusting the operating characteristics to improve the performance of an emulsified palm oil methyl ester run diesel engine*, Energy Conversion and Management **69** (2013), 191 – 198.

[59] John E. Dec, *A conceptual model of DI diesel combustion based on laser-sheet imaging*, SAE technical paper **970873** (1997).

[60] John E. Dec and Robert E. Canaan, *PLIF imaging of NO formation in a DI diesel engine*, SAE technical paper **980147** (1998).

[61] Luigi del Re, Peter Langthaler, Christian Furtmueller, Stephan Winkler, and Michael Affenzeller, *NOx virtual sensor based on*

structure identification and global optimization, SAE technichal paper **2005-01-0050** (2005).

[62] Luigi del Re, Peter Ortner, and Daniel Alberer, *Chances and challenges in automotive predictive control*, Automotive Model Predictive Control, Springer Berlin Heidelberg, 2010, pp. 1 – 22.

[63] José M. Desantes, Jean Arrègle, Santiago Molina, and Marc Lejeune, *Influence of the EGR rate, oxygen concentration and equivalent fuel/air ratio on the combustion behaviour and pollutant emissions of a heavy-duty diesel engine*, SAE technical paper **2000-01-1813** (2000).

[64] M. Diehl, H.G. Bock, H. Diedam, and P.-B. Wieber, *Fast direct multiple shooting algorithms for optimal robot control*, Fast Motions in Biomechanics and Robotics (Moritz Diehl and Katja Mombaur, eds.), Lecture Notes in Control and Information Sciences, vol. 340, Springer Berlin Heidelberg, 2006, pp. 65 – 93.

[65] Lee G. Dodge, Douglas M. Leone, David W. Naegeli, Daniel W. Dickey, and Kendall R. Swenson, *A PC-based model for predicting NOx reductions in diesel engines*, SAE technical paper **962060** (1996).

[66] Alan R. Dohner, *Optimal control solution of the automotive emission-constrained minimum fuel problem*, Automatica **17** (1981), 441 – 458.

[67] Rolf Egnell, *Combustion diagnostics by means of multizone heat release analysis and NO calculation*, SAE technical paper **981424** (1998).

[68] ———, *A simple approach to studying the relation between fuel rate, heat release rate and NO formation in diesel engines*, SAE technical paper **1999-01-3548** (1999).

[69] A. Engelsone, S.L. Campbell, and J.T. Betts, *Direct transcription solution of higher-index optimal control problems and the virtual index*, Applied Numerical Mathematics **57** (2007), no. 3, 281 – 296.

[70] Claes Ericson, Björn Westerberg, Ingemar Odenbrand, and Rolf Egnell, *Characterisation and model based optimization of a complete diesel engine/SCR system*, SAE technical paper **2009-01-0896** (2009).

[71] Fariba Fahroo and I. Michael Ross, *A spectral patching method for direct trajectory optimization*, Journal of the Astronautical Sciences **48** (2000), 269 – 286.

[72] Hans Joachim Ferreau, Peter Ortner, Peter Langthaler, Luigi del Re, and Moritz Diehl, *Predictive control of a real-world diesel engine using an extended online active set strategy*, Annual Reviews in Control **31** (2007), no. 2, 293 – 301.

[73] H.J. Ferreau, G. Lorini, and M. Diehl, *Fast nonlinear model predictive control of gasoline engines*, Proceedings of the 2006 IEEE International Conference on Control Applications, München, 2006.

[74] Martin Fink, *Automatic differentiation for Matlab*, 2007, http://www.mathworks.com/matlabcentral/fileexchange/15235.

[75] Bengt Fornberg, *Generation of finite difference formulas on arbitrarily spaced grids*, Mathematics of Computation **51** (1988), 699 – 706.

[76] Divya Garg, Michael Patterson, William W. Hager, Anil V. Rao, David A. Benson, and Geoffrey T. Huntington, *A unified framework for the numerical solution of optimal control problems using pseudospectral methods*, Automatica **46** (2010), 1843 – 1851.

[77] U. Gärtner, G. Hohenberg, H. Daudel, and H. Oelschlegel, *Development and application of a semi-empirical NOx model to various HD diesel engines*, Thermo- and Fluid Dynamic Processes in Diesel Engines 2, Selected Papers from the THIESEL 2002 Conference Valencia, Spain, 2002, pp. 285 – 312.

[78] J.A. Gatowski, E.N. Balles, K.M. Chun, F.E. Nelson, J.A. Ekchian, and J.B. Heywood, *Heat release analysis of engine pressure data*, SAE technical paper **841359** (1984).

[79] Hans P. Geering, *Optimal control with engineering applications*, Springer Berlin Heidelberg New York, 2007.

[80] J. I. Ghojel, *Review of the development and applications of the Wiebe function: A tribute to the contribution of Ivan Wiebe to engine research*, International Journal of Engine Research **11** (2010), no. 4, 297 – 312.

[81] Evangelos G. Giakoumis and Alexios I. Alafouzos, *Comparative study of turbocharged diesel engine emissions during three different transient cycles*, International Journal of Energy Research **34** (2010), 1002 – 1015.

[82] P.E. Gill, N.I. Gould, W.M. Murray, M.A. Saunders, and M.H. Wright, *Range-space methods for convex quadratic programming*, Tech. Report SOL-82-14, Systems Optimization Laboratory, Department of Operations Research, Stanford University, California, 1982.

[83] Philip E. Gill, Nicholas I. M. Gould, Walter Murray, Michael A. Saunders, and Margaret H. Wright, *A weighted gram-schmidt method for convex quadratic programming*, Mathematical Programming **30** (1984), 176 – 195.

[84] Philip E. Gill, Walter Murray, and Michael A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM Review **47** (2005), 99 – 131.

[85] Philip E. Gill and Elizabeth Wong, *Sequential quadratic programming methods*, Mixed Integer Nonlinear Programming (Jon Lee and Sven Leyffer, eds.), The IMA Volumes in Mathematics and its Applications, vol. 154, Springer New York, 2012, pp. 147 – 224.

[86] A. Griewank and Ph.L. Toint, *On the unconstrained optimization of partially separable functions*, Nonlinear Optimization (M.J.D. Powell, ed.), New York: Academic Press, 1981, pp. 301 – 312.

[87] ———, *Local convergence analysis for partitioned quasi-Newton updates*, Numerische Mathematik **39** (1982), no. 3, 429 – 448.

[88] Andreas Griewank and Andrea Walter, *Evaluating derivatives: Principles and techniques of algorithmic differentiation*, 2nd ed., SIAM, Philadelphia, 2008.

[89] Sébastien Gros, M. Zanon, and Moritz Diehl, *A relaxation strategy for the optimization of airborne wind energy systems*, European Control Conference (ECC), Zurich, 2013.

[90] C. Guardiola, J.J. Lopez, J. Martin, and D. Garcia-Sarmiento, *Semiempirical in-cylinder pressure based model for NOx prediction oriented to control applications*, Applied Thermal Engineering **31** (2011), 3275 – 3286.

[91] Mark Guerrier and Paul Cawsey, *The development of model based methodologies for gasoline IC engine calibration*, SAE technical paper **2004-01-1466** (2004).

[92] Lino Guzzella and Christopher H. Onder, *Introduction to modeling and control of internal combustion engine systems*, 2nd ed., Springer Berlin Heidelberg, 2010.

[93] Michael Hafner and Rolf Isermann, *Multiobjective optimization of feedforward control maps in engine management systems towards low consumption and low emissions*, Transactions of the Institute of Measurement and Control **25** (2003), 57 – 74.

[94] Jonathan R. Hagena, Zoran S. Filipi, and Dennis N. Assanis, *Transient diesel emissions: Analysis of engine operation during a tip-in*, SAE technical paper **2006-01-1151** (2006).

[95] Ernst Hairer and Gerhard Wanner, *Stiff differential equations solved by Radau methods*, Journal of Computational and Applied Mathematics **111** (1999), 93 – 111.

[96] J. G. Hawley, F. J. Wallace, and S. Khalil-Arya, *A fully analytical treatment of heat release in diesel engines*, Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering **217** (2003), 701 – 717.

[97] G. Heider, G. Woschni, and K. Zeilinger, *2-Zonen Rechenmodell zur Vorausrechnung der NO-Emission von Dieselmotoren*, MTZ **11** (1998), 770 – 775.

[98] Naeim A. Henein, *Analysis of pollutant formation and control and fuel economy in diesel engines*, Progress in Energy and Combustion Science **1** (1976), 165 – 207.

[99] Albert L. Herman and Bruce A. Conway, *Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules*, Journal of Guidance, Control, and Dynamics **19** (1996), 592 – 599.

[100] J. J. Hernández, M. Lapuerta, and J. Pérez-Collado, *A combustion kinetic model for estimating diesel engine NOx emissions*, Combustion Theory and Modeling **10** (2006), 639 – 657.

[101] J. J. Hernández, J. Pérez-Collado, and J. Sanz-Argent, *Role of the chemical kinetics on modeling NOx emissions in diesel engines*, Energy & Fuels **22** (2008), 262 – 272.

[102] John B. Heywood, *Internal combustion engine fundamentals*, McGraw-Hill, 1988.

[103] Alan C. Hindmarsh, Peter N. Brown, Keith E. Grant, Steven L. Lee, Radu Serban, Dan E. Shumaker, and Carol S. Woodward, *SUNDIALS: Suite of nonlinear and differential/algebraic equations solvers*, ACM Transactions on Mathematical Software **31** (2005), 363 – 396.

[104] Hiroyuki Hiroyasu, Toshikazu Kadota, and Masataka Arai, *Development and use of a spray combustion modeling to predict diesel engine efficiency and pollutant emissions, parts 1-3*, Bulletin of the JSME **26** (1983), 569 – 591.

[105] Markus Hirsch, Klaus Oppenauer, and Luigi del Re, *Dynamic engine emission models*, Automotive Model Predictive Control, Springer Berlin Heidelberg, 2010, pp. 73 – 87.

[106] Bernd Hohlbaum, *Beitrag zur rechnerischen Untersuchung der Stickstoffoxid-Bildung schnellaufender Hochleistungsdieselmotoren*, Ph.D. thesis, TH Karlsruhe, 1992.

[107] Weirong Hong, Shuqing Wang, Pu Li, Günter Wozny, and Lorenz T. Biegler, *A quasi-sequential approach to large-scale dynamic optimization problems*, AIChE Journal **52** (2006), no. 1, 255 – 268.

[108] D. T. Hountalas, N. Savva, and R. G. Papagiannakis, *Development of a new physically based semi-empirical NOx model using the measured cylinder pressure*, THIESEL 2010 Conference on Thermo- and Fluid Dynamic Processes in Diesel Engines, 2010, pp. 341 – 357.

[109] Boris Houska, Filip Logist, Moritz Diehl, and Jan Van Impe, *A tutorial on numerical methods for state and parameter estimation in nonlinear dynamic systems*, Identification for Automotive Systems (Daniel Alberer, Hakan Hjalmarsson, and Luigi del Re, eds.), Lecture Notes in Control and Information Sciences, vol. 418, Springer London, 2012, pp. 67 – 88.

[110] Matthias Ihme and Heinz Pitsch, *Modeling of radiation and nitric oxide formation in turbulent nonpremixed flames using a flamelet/progress variable formulation*, Physics of Fluids **20** (2008), 055110–1 – 055110–20.

[111] Hussein Jaddu, *Direct solution of nonlinear optimal control problems using quasilinearization and Chebyshev polynomials*, Journal of The Franklin Institute **339** (2002), 479 – 498.

[112] Mohamed Jarraya and Didier El Baz, *Implementation of distributed iterative algorithm for optimal control problems on several parallel architectures*, Journal of Systems and Software **60** (2002), no. 2, 141 – 148.

[113] Dohoy Jung and Dennis N. Assanis, *Multi-zone DI diesel spray combustion model for cycle simulation studies of engine performance and emissions*, SAE technical paper **2001-01-1246** (2001).

[114] Shivakumar Kameswaran and Lorenz T. Biegler, *Simultaneous dynamic optimization strategies: Recent advances and challenges*, Computers & Chemical Engineering **30** (2006), 1560 – 1575.

[115] Shivakumar Kameswaran and Lorenz T. Biegler, *Convergence rates for direct transcription of optimal control problems using collocation at Radau points*, Computational Optimization and Applications **41** (2008), no. 1, 81 – 126.

[116] I. M. Khan, G. Greeves, and C. H. T. Wang, *Factors affecting smoke and gaseous emissions from direct injection engines and a method of calculation*, SAE technichal paper **730169** (1973).

[117] Mayuresh V. Kothare, Peter J. Campo, Manfred Morari, and Carl N. Nett, *A unified framework for the study of anti-windup designs*, Automatica **30** (1994), no. 12, 1869 – 1883.

[118] Henrike C. Krijnsen, Wijnand E. J. van Kooten, Hans Peter A. Calis, Ruud P. Verbeek, and Cor M. van den Bleek, *Prediction of NOx emissions from a transiently operating diesel engine using an artificial network*, Chemical Engineering & Technology **22** (1999), 601 – 607.

[119] Stanislaw W. Kruczyński, *Performance and emission of CI engine fuelled with camelina sativa oil*, Energy Conversion and Management **65** (2013), 1 – 6.

[120] S. C. Kyriakides, J. C. Dent, and P. S. Mehta, *Phenomenological diesel combustion model including smoke and NO emission*, SAE technical paper **860330** (1986).

[121] F.-A. Lafossas, M. Marbaix, and P. Menegazzi, *Development and application of a 0D D.I. diesel combustion model for emissions prediction*, SAE technical paper **2007-01-1841** (2007).

[122] Magin Lapuerta, Juan José Hernández, and Octavio Armas, *Kinetic modelling of gaseous emissions in a diesel engine*, SAE technical paper **2000-01-2939** (2000).

[123] Kangyoon Lee, Maru Yoon, and Myoungho Sunwoo, *A study on pegging methods for noisy cylinder pressure signal*, Control Engineering Practice **16** (2008), 922 – 929.

[124] Oskar Leufven and Lars Eriksson, *Surge and choke capable compressor model*, Preprints of the 18th IFAC World Congress, Milano, 2011.

[125] Hu Li, Patrick Biller, Seyed Ali Hadavi, Gordon E. Andrews, Grzegorz Przybyla, and Amanda Lea-Langton, *Assessing combustion and emission performance of direct use of SVO in a diesel engine by oxygen enrichment of intake air method*, Biomass and Bioenergy **51** (2013), 43 – 52.

[126] Youcai Liang, Gequn Shu, Haiqiao Wei, and Wei Zhang, *Effect of oxygen enriched combustion and water–diesel emulsion on the performance and emissions of turbocharged diesel engine*, Energy Conversion and Management **73** (2013), 69 – 77.

[127] M. Lindgren and P.-A. Hansson, *Effects of transient conditions on exhaust emissions from two non-road diesel engines*, Biosystems Engineering **87** (2004), 57 – 66.

[128] I.S. Litvinchev, *Iterative aggregation method for constrained optimal control problems*, Journal of Computational and Applied Mathematics **39** (1992), no. 3, 315 – 328.

[129] Dong C. Liu and Jorge Nocedal, *On the limited memory BFGS method for large scale optimization*, Mathematical Programming **45** (1989), 503 – 528.

[130] Rein Luus, *Iterative dynamic programming*, Chapman & Hall / CRC, 2000.

[131] Andreas A. Malikopoulos, Dennis N. Assanis, and Panos Y. Papalambros, *Real-time self-learning optimization of diesel engine calibration*, Journal of Engineering for Gas Turbines and Power **131** (2009), 022803–1 – 022803–7.

[132] Guillaume Martin, Vincent Talon, Pascal Higelin, Alain Charlet, and Christian Caillol, *Implementing turbomachinery physics into data map-based turbocharger models*, SAE technical paper **2009-01-0310** (2009), .

[133] Benedikt Merz, *Echtzeitfähige Modellierung des dieselmotorischen Verbrennungsprozesses*, Ph.D. thesis, TH Karlsruhe, 2008.

[134] L'ubomír Miklánek, Vojtěch Klír, Miloslav Emrich, and Luděk Pohořelský, *Thermal balance method for EGR rate determination usable for real engine with uncooled EGR system*, SAE technical paper **2009-01-1114** (2009).

[135] James A. Miller and Craig T. Bowman, *Mechanism and modeling of nitrogen chemistry in combustion*, Progress in Energy and Combustion Science **15** (1989), 287 – 338.

[136] R. Miller, G. Davis, G. Lavoie, C. Newman, and T. Gardner, *A super-extended Zel'dovich mechanism for NOx modeling and engine calibration*, SAE technical paper **980781** (1998).

[137] Matthias Mrosek and Rolf Isermann, *On the parametrisation of turbocharger power and heat transfer models*, 6th IFAC Symposium on Advances in Automotive Control (AAC), München, 2010.

[138] Martin Müller, *General air fuel ratio and EGR definitions and their calculation from emissions*, SAE technical paper **2010-01-1285** (2010).

[139] Martin Müller, Elbert Hendricks, and Spencer C. Sorenson, *Mean value modelling of turbocharged spark ignition engines*, SAE technical paper **980784** (1998).

[140] Mark P. B. Musculus, *On the correlation between NOx emissions and the diesel premixed burn*, SAE technical paper **2004-01-1401** (2004).

[141] K. Narusawa, M. Odaka, N. Koike, and Y. Tsukamoto, *Study on combustion control by changing the intake charge composition of diesel engines*, JSAE Review **13** (1992), 24 – 29.

[142] Jorge Nocedal, *Large scale unconstrained optimization*, The State of the Art in Numerical Analysis, Oxford University Press, 1996, pp. 311 – 338.

[143] Jorge Nocedal and Stephen J. Wright, *Numerical optimization*, 2nd ed., Springer Science+Business Media, LLC., 2006.

[144] Engin Oezatay and Christopher Onder, *Model based sensor reconstruction of in-cylinder pressure trace using indicator cock pressure information*, CIMAC congress, Bergen, 2010.

[145] Rabih Omran, Rafic Younes, and Jean-Claude Champoussin, *Optimal control of a variable geometry turbocharged diesel engine using neural networks: Applications on the ETC test cycle*, IEEE

Transactions on Control Systems Technology **17** (2009), no. 2, 380 – 393.

[146] Rabih Omran, Rafic Younes, Jean-Claude Champoussin, Danitza Fedeli, Francois Masson, and Noureddine Guerrassi, *Genetic algorithm for dynamic calibration of engine's actuators*, SAE technical paper **2007-01-1079** (2007).

[147] Tobias Ott, Christopher Onder, and Lino Guzzella, *Hybrid-electric vehicle with natural gas-diesel engine*, Energies **6** (2013), 3571 – 3592.

[148] E.G. Pariotis, G.M. Kosmadakis, and C.D. Rakopoulos, *Comparative analysis of three simulation models applied on a motored internal combustion engine*, Energy Conversion and Management **60** (2012), 45 – 55.

[149] Michael A. Patterson and Anil V. Rao, *Exploiting sparsity in direct collocation pseudospectral methods for solving optimal control problems*, Journal of Spacecraft and Rockets **49** (2012), 364 – 377.

[150] F. Payri, J.M. Luján, J. Martín, and A. Abbad, *Digital signal processing of in-cylinder pressure for combustion diagnosis of internal combustion engines*, Mechanical Systems and Signal Processing **24** (2010), no. 6, 1767 – 1784.

[151] F. Payri, J.R. Serrano, P. Fajardo, M.A. Reyes-Belmonte, and R. Gozalbo-Belles, *A physically based methodology to extrapolate performance maps of radial turbines*, Energy Conversion and Management **55** (2012), 149 – 163.

[152] S. L. Plee, T. Ahmad, and J. P. Myers, *Flame temperature correlation for the effects of exhaust gas recirculation on diesel particulate and NOx emissions*, SAE technical paper **811195** (1981).

[153] Warren B. Powell, *Approximate dynamic programming*, John Wiley & Sons, Inc., 2007.

[154] M. A. Psota and A. M. Mellor, *Dynamic application of a skeletal mechanism for DI diesel NOx emissions*, SAE technical paper **2001-01-1984** (2001).

[155] Carole Quérel, Olivier Grondin, and Christophe Letellier, *State of the art and analysis of control oriented NOx models*, SAE technical paper **2012-01-0723** (2012).

[156] C. D. Rakopoulos, E. G. Giakoumis, and D. C. Rakopoulos, *Cylinder wall temperature effects on the transient performance of a turbocharged diesel engine*, Energy Conversion and Management **45** (2004), 2627 – 2638.

[157] C.D. Rakopoulos and E.G. Giakoumis, *Diesel engine transient operation*, Springer London, 2009.

[158] Louis B. Rall, *Automatic differentiation: Techniques and applications*, Lecture notes in computer science, vol. 120, Springer, Berlin, 1981.

[159] Anil V. Rao, *A survey of numerical methods for optimal control*, Advances in the Astronautical Sciences, vol. 135, 2009, pp. 497 – 528.

[160] Siegfried M. Rump, *Verification methods: Rigorous results using floating-point arithmetic*, Acta Numerica **19** (2010), 287 – 449.

[161] R. W. H. Sargent, *Optimal control*, Journal of Computational and Applied Mathematics **124** (2000), 361 – 371.

[162] Alexander Schilling, Alois Amstutz, Christopher H. Onder, and Lino Guzzella, *A real-time model for the prediction of the NOx emissions in DI diesel engines*, Proceedings of the 2006 IEEE International Conference on Control Applications, Munich, 2006, pp. 2042 – 2047.

[163] Adam Lowell Schwartz, *Theory and implementation of numerical methods based on Runge-Kutta integration for solving optimal control problems*, Ph.D. thesis, University of California, Berkeley, 1996.

[164] Heiko Sequenz and Rolf Isermann, *Emission model structures for an implementation on engine control units*, Preprints of the 18th IFAC World Congress, Milano, 2011, pp. 11851 – 11856.

[165] J.R. Serrano, F.J. Arnau, V. Dolz, A. Tiseira, and C. Cervelló, *A model of turbocharger radial turbines appropriate to be used in zero- and one-dimensional gas dynamics codes for internal combustion engines modelling*, Energy Conversion and Management **49** (2008), no. 12, 3729 – 3745.

[166] Sameh Shaaban, *Experimental investigation and extended simulation of turbocharger non-adiabatic performance*, Ph.D. thesis, Universität Hannover, 2004.

[167] S. M. Shahed, W. S. Chiu, and V. S. Yumlu, *A preliminary model for the formation of nitric oxide in direct injection diesel engines and its application in parametric studies*, SAE technical paper **730083** (1973).

[168] Seha Son and Arthur E. Kolasa, *Estimating actual exhaust gas temperature from raw thermocouple measurements acquired dur-*

ing transient and steady state engine dynamomenter tests, SAE technical paper **2007-01-0335** (2007).

[169] Liviu-Constantin Stan and Daniela-Elena Mitu, *Simplified mechanism used to estimate the NOx emission of diesel engine*, Proceedings of the 2nd International Conference on Manufacturing Engineering, Quality and Production Systems, Romania, 2010, pp. 61 – 64.

[170] Gunnar Stiesch and Günter P. Merker, *A phenomenological model for accurate and time efficient prediction of heat release and exhaust emissions in direct-injection diesel engines*, SAE technical paper **1999-01-1535** (1999).

[171] D. Tieu, W.R. Cluett, and A. Penlidis, *A comparison of collocation methods for solving dynamic optimization problems*, Computers and Chemical Engineering **19** (1995), 375 – 381.

[172] D.J. Timoney, J.M. Desantes, L. Hernandez, and C.M. Lyons, *The development of a semi-empirical model for rapid NOx concentration evaluation using measured in-cylinder pressure in diesel engines*, Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering **219** (2005), 621 – 632.

[173] Michael L. Traver, Richard J. Atkinson, and Christopher M. Atkinson, *Neural network-based diesel engine emissions prediction using in-cylinder combustion pressure*, SAE technical paper **1999-01-1532** (1999).

[174] Yasunori Urano, Yutaka Nakano, Hitoshi Takada, and Mitsuharu Sugita, *Optimization technique for transient emission reduction of heavy duty diesel engine*, SAE technical paper **2005-01-1099** (2005).

[175] Sergey Ushakov, Harald Valland, and Vilmar Æsøy, *Combustion and emissions characteristics of fish oil fuel in a heavy-duty diesel engine*, Energy Conversion and Management **65** (2013), 228 – 238.

[176] Greg von Winckel, *Finite difference weights*, 2004, http://www.mathworks.com/matlabcentral/fileexchange/5269.

[177] Andreas Wächter and Lorenz T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Mathematical Programming **106** (2006), 25 – 57.

[178] W. Wagner and A. Pruss, *International equations for the saturation properties of ordinary water substance. Revised according to the international temperature scale of 1990*, Journal of Physical and Chemical Reference Data **22** (1993), 783 – 787.

[179] Johan Wahlström, Lars Eriksson, and Lars Nielsen, *Controller tuning based on transient selection and optimization for a diesel engine with EGR and VGT*, SAE technical paper **2008-01-0985** (2008).

[180] F. J. Wallace, S. Khalil-Arya, J. G. Hawley, and D. Dang, *A zero dimensional simulation model for HSDI diesel engines for predicting NOx, spray and swirl formation*, Thermo- and Fluid Dynamic Processes in Diesel Engines, Selected Papers from the THIESEL 2002 Conference Valencia, Spain, 2002, pp. 363 – 381.

[181] Lin Wang, Hui Peng, Hua-Yong Zhu, and Lin-Cheng Shen, *A survey of approximate dynamic programming*, International Conference on Intelligent Human-Machine Systems and Cybernetics **2** (2009), 396 – 399.

[182] Marco Warth, Peter Obrecht, Andrea Bertola, and Konstantinos Boulouchos, *Predictive phenomenological C.I. combustion modeling optimization on the basis of bio-inspired algorithms*, SAE technical paper **2005-01-1119** (2005).

[183] N. Watson and M. S. Janota, *Turbocharging the internal combustion engine*, The Macmillan Press Ltd, London and Basingstoke, 1982.

[184] German Andreas Weisser, *Modelling of combustion and nitric oxide formation for medium-speed DI diesel engines: A comparative evaluation of zero- and three-dimensional approaches*, Ph.D. thesis, ETH Zurich, 2001.

[185] Sebastian Paul Wenzel, *Modellierung der Russ- und NOx-Emissionen des Dieselmotors*, Ph.D. thesis, Otto-von-Guericke-Universität, Magdeburg, 2006.

[186] Anders Westlund and Hans-Erik Angström, *Fast physical emission predictions for off-line calibration of transient control strategies*, SAE technical paper **2009-01-1778** (2009).

[187] Carl Wilhelmsson, Per Tunestål, Bengt Johansson, Anders Widd, and Rolf Johansson, *A physical two-zone NOx model intended for embedded implementation*, SAE technical paper **2009-01-1509** (2009).

[188] Robert C. Yu and Syed M. Shahed, *Effects of injection timing and exhaust gas recirculation on emissions from a D.I. diesel engine*, SAE technical paper **811234** (1981).

[189] Jincai Zheng and Quianfan Xin, *Theoretical analysis of diesel engine NOx and soot with heuristic macro-parameter-dependent*

approach and virtual multi-zone real-time models, SAE technical paper **2009-01-2836** (2009).

Curriculum vitae

Personal data

Name	Jonas Asprion
Date of birth	April 17, 1984
Place of birth	Kilchberg ZH, Switzerland
Citizen of	Escholzmatt LU, Switzerland

Education

2009 – 2013	PhD student and teaching assistant *Institute for Dynamic Systems and Control* *ETH Zurich, Switzerland*
2007 – 2009	MSc in Mechanical Engineering *ETH Zurich, Switzerland*
2004 – 2007	BSc in Mechanical Engineering *ETH Zurich, Switzerland*
1997 – 2003	Eidgenössische Matura, Typus A *Stiftsschule Einsiedeln, Switzerland*

Publications

The publications are listed in the Bibliography and comprise the Journal Papers [18, 19, 20, 11] and the Conference Contributions [22, 21, 17].