Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Control Engineering

Distributed Identification of Nonlinear Systems using Regularization

by
Radek Beňo

Presented to the Faculty of Electrical Engineering,
Czech Technical University in Prague,
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Ph.D. Program: Electrical Engineering and Information Technology
Branch of Study: Control Engineering and Robotics
Supervisor: prof. Ing. Vladimír Havlena, CSc.
Prague, February 2018

Supervisor:
prof. Ing. Vladimír Havlena, CSc.
Department of Control Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2
166 27 Prague 6
Czech Republic

Supervisor specialist:
Ing. Daniel Pachner, PhD.
Honeywell Prague Laboratory
V Parku 2326/18
148 00 Prague 4
Czech Republic

To my family

Declaration

This doctoral thesis is submitted in partial fulfillment of the requirements for the degree of doctor (Ph.D.). The work submitted in this dissertation is the result of my own investigation, except where otherwise stated. I declare that I worked out this thesis independently and I quoted all used sources of information in accord with methodical instructions about ethical principles for writing an academic thesis. Moreover, I declare that it has not already been accepted for any degree and is also not being concurrently submitted for any other degree.

Prague, February 2018

Radek Beňo

Acknowledgements

First of all, I would like to thank my dissertation thesis supervisor, prof. Ing. Vladimír Havlena, CSc., for his guidance and support.

I would also like to thank my supervisor specialist, Ing. Daniel Pachner, PhD., for his guidance and willingness to help me throughout my entire studies and especially for his endless patience, especially when anything was not going well.

Finally, I would like to express my gratitude to my family for supporting me during the whole time of my studies.

The research in this thesis has been supported by the Grant Agency of the Czech Republic, grant P103/11/1353.

Thesis advisor: prof. Ing. Vladimír Havlena, CSc. Radek Beňo

Distributed Identification of Nonlinear Systems using Regularization

Abstract

This thesis deals with a new method of identifying nonlinear systems which consist of subsystems called components. The identification problem is here understood as a process of parameters' calibration of nonlinear systems with fixed structure. The whole work deals with the calibration of parameters of nonlinear systems in steady states. One of the greatest contribution of the work is the component regularization methodology, which mainly brings better numerical stability of the solution.

The presented algorithm is distributed and decomposes the original problem according to the primal decomposition to a series of simpler subproblems, in which one particular steady state of the system is solved, i.e. fitted to the data, according to a given global parameter vector. These subproblems can be solved independently of each other, a global optimizer collects these individual contributions and iteratively changes the parameter's values according to an optimization criterion.

Regularized components support the calibration process in particular by correct definition of the domain of the model's validity, i.e. the area where the model is numerically well conditioned, and numerical stability is further strengthened by introduction of additional variables that constrain the input, output and internal signals of components. These additional constraints limit the propagation of nonphysical signal values across the entire system model.

A Mean-Value Model has been chosen as a type of system model over which distributed optimization works, which makes possible not only to model the basic physical phenomena of the system, but also to use it well within the framework of the further system control design.

The presented method is demonstrated in this thesis on a specific example of the calibration of the non-linear model of a Diesel internal combustion engine.

Keywords: Distributed Optimization, System Identification, Nonlinear Systems, Parameter Calibration

Vedoucí práce: prof. Ing. Vladimír Havlena, CSc. Radek Beňo

Distribuovaná identifikace nelineárních systémů s využitím regularizace

Abstrakt

Tato práce se zabývá novou metodou identifikace nelineárních systémů, přičemž největším přínosem práce je rozpracování metodiky tzv. regularizace komponent, která přináší především lepší numerickou stabilitu řešené úlohy. Úloha identifikace je zde chápána jako metoda kalibrace parametrů nelineárních systémů při jejich pevně zvolené struktuře. Celá práce se věnuje kalibraci parametrů nelineárních systémů v ustálených stavech.

Prezentovaný algoritmus je distribuovaný a rozkládá původní úlohu podle primární dekompozice na řadu jednodušších podúloh, ve kterých je řešen vždy jeden konkrétní ustálený stav soustavy při daném globálním vektoru parametrů. Tyto podúlohy lze řešit nezávisle na sobě, přičemž po vyřešení shromažďuje jednotlivé příspěvky globální optimalizátor, který iterativně mění vektor parametrů v souladu se svým optimalizačním kritériem.

Regularizované komponenty podporují kalibrační proces zejména správným vymezením oblasti platnosti modelu, tj. oblasti, kde je model dobře numericky podmíněn, a tuto numerickou stabilitu dále podporují pomocí zavedených dodatečných proměnných, které udržují vstupní, výstupní, ale i vnitřní signály komponent v mezích, které značně omezují šíření nefyzikálních hodnot signálů skrze celý model soustavy.

Jako model soustavy, nad kterým distribuovaná optimalizace pracuje, byl zvolen tzv. model středních hodnot (Mean-Value Model), díky kterému lze nejenom dobře uchopit základní fyzikální jevy soustavy, ale je dobře použitelný i v rámci dalšího návrhu řízení soustavy.

Představená metoda je v rámci práce demostrována na konkrétním příkladu kalibrace nelineárního modelu dieslového spalovacího motoru.

Klíčová slova: Distribuovaná optimalizace, identifikace systémů, nelineární systémy, kalibrace parametrů

Contents

1 Introduction 1

1.1 Motivation 3

1.2 Problem Statement 6

1.3 Goals of the Dissertation Thesis 7

1.4 Structure of the Dissertation Thesis 8

2 Nonlinear System Modelling and Calibration 11

2.1 Introduction 11

2.2 Nonlinear Systems 12

2.3 System Calibration Methods 13

2.4 Shooting Methods 18

2.5 Distributed Optimization Techniques 21

2.6 Distributed Constrained Least Squares Algorithm 26

3 Combustion Engine Modelling 33

3.1 System Overview 34

3.2 Common Modelling Principles 38

3.3 Mean Value Model of Internal Combustion Engine 40

3.4 Input/Output System Review 48

4 Regularized Component Concept 51

4.1 Problems of Steady-State Calibration 52

4.2 Regularized Component Concept 53

4.3 Model Regularization Example 57

5 The Algorithm 61
5.1 Constrained Least Squares for Distributed Calibration 61
5.2 Gauss-Newton Method 63
5.3 Algorithm Applications Examples 67

6 Mean Value Model of Internal Combustion Engine 75
6.1 Turbocharger Nonlinearity 75
6.2 Compressor Singularities 77
6.3 Turbine 78
6.4 Flow Restrictions 78
6.5 Recirculation Loops 79

7 Engine Model Calibration 81
7.1 Compressor 82
7.2 Heat Exchangers 83
7.3 Combustion Model 84
7.4 Turbine 85
7.5 Model Constraints 86
7.6 Model Calibration 86
7.7 Calibration Results 90
7.8 After the Steady-State Calibration 90

8 Conclusion 93
8.1 Summary 93
8.2 Contributions of the Author 95
8.3 Open Problems 98
8.4 Fulfillment of the Stated Goals 98

References 104

List of figures

2.2.1 Process of steady state data collection. 13

2.3.1 Geometrical representation of fundamental constrained optimization problem. For the sake of simplicity, there is no equality constraint. 14

2.4.1 Typical run of the single-shooting algorithm. 20

2.4.2 Graphical interpretation of the multiple-shooting method. Figure was inspired by [29]. 21

2.5.1 Graphical representation of Primal and Dual decompositions. . . 26

2.6.1 Graphical interpretation of the linear dependence of parameters on a local variable. 30

3.1.1 Typical structure of Diesel internal combustion engine. 35

3.1.2 Structure of diesel turbocharger [3]. 35

3.1.3 4 stroke diesel engine intercooler [1]. 37

3.1.4 Typical valve body [2]. 38

3.3.1 Typical parameters fitting from the compressor map. 44

3.3.2 Fitting of the turbine model parameters. Figure was taken from lectures of prof. Ilja Kolmanovsky. 45

3.4.1 Input/output ICE model review according to a schema presented in Fig. 3.1.1. 50

4.1.1 Steady-state model of component-grey part represents model nonlinear equations. Without the application of internal variables, the structure can not be changed, the simulation and calibration process is not separated in any way. 53
4.1.2 Steady-state component model. The signal goes through the numerically safe and numerically unsafe parts of the component. Here, for sake of clarity, numerically unstable parts are separated into unsafe input part related to maintain the span of input signals at a physically relevant intervals and into the intern unsafe part which is related to the need of preserve the physical relevance of the component model. 54
4.2.1 Steady-state model component with regularization. Grey part represents model nonlinear equations, yellow part represents nonlinear equality constraints and blue part represents linear equality constraints. Variable flag 0,1 switch the component between simulation (classical component approach = 0) and calibration (regularized component approach = 1) mode. 56
4.2.2 Steady-state model component with regularization. 57

5.1.1 Model structure modified with slack variables. 63
5.3.1 Calibrated cascade of the affine systems. 68
5.3.2 Calibration results of the cascade of affine systems. 69
5.3.3 Calibrated nonlinear system. 70
5.3.4 Affine and nonlinear system calibration result. 71
5.3.5 Affine and nonlinear system calibration result. 72
5.3.6 Result of the test of the computational speed. 73
5.3.7 Result of the test of the computational speed. 74

7.0.1 Turbocharged diesel engine model schematics. 82
7.1.1 Compressor flow map and its fit by rational polynomial function. 83
7.1.2 Compressor efficiency map and its fit by rational polynomial function. 84
7.6.1 Turbine speed steady-state identification results. 87
7.6.2 Fresh air flow steady-state identification results. 88

7.6.3 Pressure before turbine steady-state identification results. 88
7.6.4 Intake manifold pressure steady-state identification results. . . . 89
7.8.1 Transient response of the MVM to EGR valve position step from 20 % to 40 %. 92

8.1.1 Graphic interpretation of calibration results before and after running global level calibration from Table 7.7.1. Yellow bars are related to fit quality before global level calibration and blue bars after it. 96

List of Tables

2.6.1 Description of the linear dependence related to the Figure 2.6.1. 31

3.4.1 Independent variables and the corresponding constraints contained in the Figure 3.4.1. 48

5.3.1 Result of parameters values within the cascade of affine system calibration. 69
5.3.2 Result of parameters values within the nonlinear system calibration. 71
5.3.3 Problem assignment for computational speed test. 73

7.7.1 Calibration results before and after running global level calibration. 91

Nomenclature

Acronyms and abbreviations

SS	Single Shooting
MS	Multiple Shooting
QP	Quadratic Programming
CFD	Computational Fluid Dynamics
ICE	Internal Combustion Engine
EGR	Exhaust Gas Recirculation
DPF	Diesel Particulate Filter
SCR	Selective Catalytic Reduction
COM	Control Oriented Model
LNT	Lean NOx Trap
MVM	Mean Value Model
VGT	Variable Geometry Turbine

Used physical variables and its dimensions

	Flows	
$\dot{m}$	Component mass flow	kg/s
$\dot{m}_{\mathrm{ref}}$	Reference mass flow	kg/s
$\dot{m}_{\mathrm{i}}$	Component inlet mass flow	kg/s
$\dot{m}_{\mathrm{o}}$	Component outlet mass flow	kg/s
$\dot{m}_{\mathrm{c}}$	Compressor mass flow	kg/s
$\dot{m}_{\mathrm{t}}$	Turbine mass flow	kg/s
$\dot{m}_{\mathrm{eng}}$	Engine mass flow	kg/s
$\dot{m}_{\mathrm{fuel}}$	Fuel mass flow	kg/s
m_{inj}	Injected quantity	mg
	Volumes	
V_{d}	Displacement volume (all cylinders)	m^3
	Temperatures	
T	Temperature	K
T_{ref}	Reference temperature	K
T_{i}	Component inlet temperature	K
T_{o}	Component outlet temperature	K
T_{cm}	Cooling medium temperature	K
T_1	Compressor inlet temperature	K
T_2	Intake manifold temperature	K
T_3	Exhaust manifold temperature	K
T_4	Turbine outlet temperature	K
	Pressures	
p_{ref}	Reference pressure	kPa
p_{amb}	Ambient pressure	kPa
p_{i}	Component inlet pressure	kPa
p_{o}	Component outlet pressure	kPa
p_1	Compressor inlet pressure	kPa
p_2	Intake manifold pressure	kPa
p_3	Exhaust manifold pressure	kPa
p_4	Turbine outlet pressure	kPa
	Concentrations	
$[O_2]_{\mathrm{i}}$	Inlet oxygen concentration	kg/kg
$[O_2]_{\mathrm{o}}$	Outlet oxygen concentration	kg/kg
ξ	Oxygen-fuel stoichiometric ratio	-

	Densities	
ρ_1	Compressor inlet fluid density	kg/m^3
ρ_2	Intake manifold fluid density	kg/m^3
ρ_{i}	Inlet gas density	kg/m^3
	Other variables and parameters	
d_{c}	Compressor diameter	m
N_{e}	Engine rotational speed	rpm
κ	Turbine flow parameter	-
T_s	Sampling period	s
Ω	Invariant set	-
c_{v}	Specific heat under constant volume	J / kg / K
c_{p}	Specific heat under constant pressure	J / kg / K
N_{tc}	Turbocharger rotational speed	rpm
P_{c}	Compressor power	kW
η_{c}	Compressor isentropic efficiency	-
η_{he}	Heat exchanger effectiveness	-
η_{vol}	Volumetric efficiency	-
Φ_{c}	Dimensionless flow rate	-
Ψ_{c}	Dimensionless head parameter	-
a	Heat exchanger parameter	kg/s
k_i	Experimental constant	-
γ	Specific heat ratio	-
Ψ	Dimensionless pressure ratio	-
T_{q}	Torque	Nm
H_{LV}	Fuel heating value	J/mg
u_{VGT}	Vane position (map)	-
$\tilde{u}_{\mathrm{VGT}}$	Vane control signal	%
c_{d}	Discharge coefficient	-
A	Flow restriction area	m^2
U_{c}	Blade tip speed	m/s
M	Inlet Mach number	-
	Mathematical symbols	
ε	Small positive number	-
O	Zero matrix	-
I	Identity matrix	-
o	Zero vector	-
R^2	$R^2 = 1 - \frac{\sum_k (y_k - g(x_k,u_k,p))^2}{\sum_k (y_k - \bar{y})^2}$	-

In the beginning there was nothing, which exploded.

Terry Pratchett

1 Introduction

Optimization is all around us. We meet it in everyday life, whether in the form of devices that help us solve common problems, such as how quickly and at what cost we get by the car to the place we want, or in the form of optimization tasks that stand before us, from what to wear due to today's weather, deciding which product has a better price/performance ratio, and deciding on who to vote for the upcoming elections.

Even modern society itself exerts a certain pressure for things or problems to be solved optimally, that is, in accordance with many inexhaustible sets of criteria to reflect a given view on the right solution to the problem. And so we are talking about an optimal state budget, choosing optimal product marketing strategies, or optimizing the control of a Diesel combustion engine. All these problems can be captured in a certain way by means of mathematical equations or physical principles, thus creating a mathematical model that is then a simplification of the perception of a real problem or system. With this inaccurate representation it is possible to precisely apply the existing optimization methods, find the desired solution and apply it back to the real world.

Nowadays, it is no longer a problem to model the behavior of simple systems on the basis of physics, which can be optimally controlled in the context of contemporary optimization methods. The practice is that in a given system only the most important phenomena are modelled based on physical principles so that the resulting model is not too complex, but that at the same time it reflects the behavior that we want to model. The model is specified by its parameters – for example, the model of the inverted pendulum describes the mass balance behavior with the unstable equilibrium, but parameters such as weight or position of center of mass determine whether it is a broom on the fingers of your hand, a Segway or a spacecraft. Some model parameters can be measured directly, others need to be obtained using so-called identification from experimental data measured on the real system. These data carry information on the behavior of the system, i.e. how the system responds to the change of input by changing its output. After the identification phase, the model is already ready and can be used to design the system's control strategy. In other words, the better the model of the system (the more accurately it represents the mathematical form of the behavior of the real system), the better the control of this system can be further designed. The big advantage of this approach is that there is no need to test different management strategies on a real system, for example, where it risks destruction, or where the experiment can not be done with the control system – for example, nuclear power plant or airliner control. In the context of modern control theory, the model and its precision are especially preferred in the Model Predictive Control (MPC). Here, based on the model, (its present state and output), the future output of the real system is predicted and the future input to the system is optimized so that its control meets the requirements. Any inaccuracy of the model then manifests itself more the farther into the future we want to predict its behavior in the future. However, it is clear that the modern systems we want to model and subsequently control are not simple and often consist of a large number of components that interact with one another. The first idea that comes to mind, of identifying all the components of the system, and then putting them together in the whole system, appears to be incorrect, because in such structured systems, (as in the case of the human being), applies the so-called holistic principle, which states that the system is more than just the joining of individual components. In other words, locally correct component models may not describe the correct behavior of the system as a whole. Thus, there emerges a need to identify the system as a whole.

This dissertation deals with a very small section of all the above mentioned. An optimization method of identifying model parameters for complex models consisting of components that are modelled from basic physical principles is presented here. First, however, it is necessary to modify all the components so that there appear no numerical problems that could make difficult the whole identification process. At the same time, this is also a protection against propagation of a systemic error, where for example, a component can give inaccurate data on its output due to its wrong parameters, which in turn can be input data for the next component. In this work, this adjustment is called regularization.

1.1 Motivation

In the early days of the control theory, the controlled system was not being identified and control in most cases in the form of PID (proportional-integral-derivative) or other controllers were mostly designed directly on the controlled system. Over time, controlled systems have become more complex which lead to greater requirements on control methods, which began tuning of controllers using the system model. As in the case of controllers, there are plenty of literature that deal with practical advice on how to identify simple models. However, these tips stop working as soon as the model becomes too complex in structure and contains lots of parameters that need to be calibrated.

With the introduction of modern control theory, which introduces a very important concept of optimal control, the optimization methods have begun to be used to identify systems. It is the use of optimization methods that brought great comfort in the fact that we get the best possible solution, the best possible model, in accordance with the chosen criterion. In this thesis, the problem of system identification, or better, the calibration of system parameters, is understood as an optimization task.

In the identification of dynamic systems using optimization, there is currently a large number of methods that accurately solve predefined problems, see [5], [16], etc). The main source of problems, therefore, is the part where the well-described identification theory begins to encounter a concrete, practical part of its use – a model of a system that tries to best reflect the reality it describes. The Diesel combustion engine model, which is used in this work as a sample example of a complex non-linear identified system, is the source of many such problems.

The models often combine empirical, mechanical and thermodynamical laws and chemical kinetics. They range from simple local linear models to complex Computational Fluid Dynamics (CFD) models. The high fidelity first-principles models are used mainly for off-line analysis, system optimization and diagnostics design. Beside the dynamic models, the global steady-state nonlinear high fidelity models are often used in set-point optimization. In contrast, the real-time feedback control design is still predominantly not model based, or it is based on local linear models. The reason for that is that it is difficult to develop a first-principles-based model with sufficient accuracy. However, such models would have a clear advantage of global validity and better extrapolation capabilities.

The Mean Value Models (MVM) represent an example of such first-principles models potentially useful for model-based controls, see [18], [22] and [32]. Especially in the automotive industry there are "zero dimensional" models which consider the average mass and energy flows over the engine cycles, neglecting their pulsations caused by the periodic emptying and filling of cylinders [20]. These models are primarily used for design of air path controls. The model is built mainly around the differential equations of the gas pressure and temperature at certain control volumes, where the pressure and temperature is assumed to be constant.

In automotive industry, it is a common practice that physical models are calibrated component-wise. This presents a different situation than the process industry, where ingenious nonlinear model identification methods were developed [27]. A typical engine test cell is equipped with a sufficient number of sensors which provide input and output signals to component sub-models. This makes the model calibration problem a static fitting problem; the individual nonlinearities are fitted separately. Such calibration approach is simpler compared to the general nonlinear dynamic model identification.

The minimization of prediction errors on the component level does not guarantee the minimum prediction error of the resulting dynamic model. Accuracy of the model built from separately fitted components may be sub-optimal. There are two reasons for this. Firstly, the model structure is imperfect. The fitted components models may require some adjustment to compensate for the effect of the structural simplifications. Secondly, the turbocharged ICE represents a feedback structure where the component errors are propagated and possibly amplified. It is thus necessary to minimize the errors which are amplified the most even at the cost of making the component fit worse on a local level.

This is why a system level optimization based calibration approach has been proposed [33]. The idea is to start from the component level model and run an optimization of model parameters to fit the global model predictions to the data. This automated model calibration is still relatively new in automotive industry and models are often adjusted manually based on physical insight. The reason is that the optimization can drive model parameters to incorrect values or values which make some of the internal signals physically incorrect albeit the prediction errors of the optimized signals are minimized. This could be improved by constraining model parameters during the optimization to certain a priori described sets, regularizing the problem enforcing the prior information about model components [40]. It is also important to fit all available measurements during the automated calibration. The automotive engineering community often regards such optimized models with some suspicion and their predictions are considered less reliable compared to the models built from components.

At the same time, the MVM of turbocharged engines contain nonlinear functions with singularities and constrained domains. In [23], it has been proven that the states of this physical system remain in certain invariant set Ω within the singularities and argument of functions remain in their domains. An accurate simulation of a properly calibrated model started in Ω should stay there. Nonetheless, a numerical simulation of the model can fail if the solution will fall outside Ω due to discretization errors; crossing the singularity. System level calibration of such a model is difficult. As the set Ω can depend on model parameters, the optimization can hit infeasible signal values when optimizing the parameters which makes the numerical properties of the optimization problematic. The model Jacobians can be ill-conditioned close to singularities, models can be unstable or even have finite escape time outside Ω. How system level model calibration can be approached in such a situation is studied in this thesis. The idea is to use constraints on both parameters and model internal signals as a regularization of the optimization process to prevent the optimization from exploring infeasible areas.

The model singularities also affect the numerical solution of the differential equations. Close to a singularity, or to a point where the right-hand side of the differential equation is not differentiable, the model Jacobian is ill-conditioned which is a manifestation of model stiffness [19]. The numerical solution may then be difficult and special implicit solvers are required. At a singularity, the differential equations are even not guaranteed to have a unique solution (Picard Lipschitz Theorem). A typical example of a point where the right-hand side of the model differential equation is not differentiable is a model with the valve flow equation when the pressure ratio across the valve is one. A regularization of the valve flow function for such pressure ratios is discussed in [18]. There has been shown how a smooth polynomial approximation can replace the non-differentiable function close to the point of non-differentiability. The model differential equations treated in this way will be less stiff, and will definitely be Picard Lipschitz, which means easier numerical solution. The approach proposed by this thesis is different as it is steady-state only. It constrains the model signals to be always in certain ε distance from the singularities. The model is not allowed to enter “forbidden ground” during the steady-state calibration. This avoids not only singularities but also other undesired or physically implausible signal values. Very often slow and problematic model simulations are avoided.

1.2 Problem Statement

The nonlinear continuous-time dynamical model is defined in the usual form [25]:

$$\mathrm{d}\mathbf{x}(t)/\mathrm{d}t = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}), \tag{1.1}$$

$$\mathbf{y}(t) = g(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}). \tag{1.2}$$

The model calibration problem is formulated as a deterministic nonlinear least squares optimization solved with respect to vector of model parameters $\mathbf{p}$ considering measured sampled sequences of model input and output vectors, $\mathbf{u}(k)$ and $\mathbf{y}(k)$ respectively; $k = 1 \ldots K$. Here $\mathbf{y}(k)$ and $\mathbf{u}(k)$ denote signal values of the continuous time signals $\mathbf{y}(t)$ and $\mathbf{u}(t)$ sampled at discrete sampling instants $t_k = \lim_{\varepsilon \to 0^+}(kT_s - \varepsilon)$; with sampling period T_s. The minimized sum of prediction error squares will be referred as the cost function

$$\begin{aligned} \hat{\mathbf{p}} = \arg\min_{\mathbf{p},\mathbf{x}_0} \sum_{k=1}^{K} & \quad \|g(\mathbf{x}(k), \mathbf{u}(k), \mathbf{p}) - \mathbf{y}(k)\|_2^2\,, \\ \text{subj. to} & \quad f(\mathbf{x}(k), \mathbf{u}(k), \mathbf{p}) = 0. \end{aligned} \tag{1.3}$$

Here g is the output function of (1.2), $\hat{\mathbf{p}}$ denotes the vector of parameter estimates and equality constraint $f(\mathbf{x}(k), \mathbf{u}(k), \mathbf{p}) = 0$ determines that the problem will be solved only in the steady states of the model equation f.

The method to solve this nonlinear optimization is always based on solving the model equations iteratively. In many cases, the defined optimization problem is too complex for a straightforward solution, and to solve it in a reasonable time, the original problem needs to be transformed into a distributed form. Solution with the current parameter estimates and current initial conditions estimates provides the nominal trajectory as well as its sensitivity with respect to the optimized variables. This defines the cost function value and its gradient. These values are supplied to a convex optimization method, e.g. Gauss-Newton (see section 2.3.2), equipped with a suitable step control technique such as Levenberg-Marquardt (see section 2.3.3) or trust region methods.

1.3 Goals of the Dissertation Thesis

The main goals of this dissertation thesis are as follows:

1. Study numerical problems that occur during the calibration of nonlinear systems.

2. Analyze regularization techniques which can improve a numerical instability of a calibration process.

3. Find a solution of these numerical problems using regularization techniques and develop a methodology for its usage.

4. Test a designed solution on a chosen nonlinear model.

1.4 Structure of the Dissertation Thesis

This thesis is organized into four main parts. The first part consists of an introduction to nonlinear system modelling and calibration and modelling of internal combustion engines. Secondly, the concept of regularized component is introduced. Third part focuses on distributed calibration algorithm and the fourth part is dedicated to the regularized model of combustion engine and its calibration.

1.4.1 Chapter 1

Chapter 1 describes the motivation and goals of the thesis.

1.4.2 Chapter 2

In this chapter, the fundamentals of nonlinear system identification and calibration are given. An overview of the different methods typically used for calibration of these systems with focus on steady-state calibration and methods of distributed optimization are also described.

1.4.3 Chapter 3

The most common techniques of air path of diesel engine modelling are presented. There is also given an overview of models used in following chapters.

1.4.4 Chapter 4

The concept of regularized component is introduced.

1.4.5 Chapter 5

This chapter focuses on distributed calibration algorithm and some examples of its use are given.

1.4.6 Chapter 6

The full model of air path of internal combustion engine with usage of regularized components is presented in this chapter.

1.4.7 Chapter 7

In this chapter, the calibration of an Internal Combustion Engine model is shown.

1.4.8 Chapter 8

At the end the conclusion of the whole work is given.

Some parts of this thesis are build on the results that were previously published in collaboration with colleagues. Specifically, the main parts of Chapter 5, Chapter 6 and 7 were presented in [6].

If I have seen further it is by standing on the shoulders of Giants.

Isaac Newton

2

Nonlinear System Modelling and Calibration

2.1 Introduction

One of the most important tasks of current technologies is to solve optimization problems in order to suit given criteria. However, this is often an extremely complicated problem, because current systems are complex not only in the number of inputs or states, but also in the number of components they are composed of. The optimality of the solution depends on the criterion of convenience – the cost function.

Within the system calibration, the structure of the system is predetermined and is generally described in the system model using nonlinear differential equations that reflect the general behavior of the system. In order for the model to behave in accordance with the particular system, it is necessary to determine the parameters of the system that specify the general behavior affected by the differential equations.

Therefore, it is necessary that the given cost function has to be looked for so that the parameters of the system are contained in the optimization variable's place. As a rule, during optimization, these parameters are changed so that the model responds with the same outputs to the same inputs as the real system. Each method has its own way of changing parameters – parameters can be changed, for example, randomly, as in Monte Carlo methods use [35], or systematically.

From the systematic parameter changing point of view, one of the leading methods is a linearization of governing equations that has both its pros and cons. One of the biggest advantages of linearization is the fact that the principle of superposition begins to apply, and the resulting solutions can be composed. Among the big disadvantages is the fact that if linearization is done along the solution of the system, it is not possible to deviate too much from this linearization precisely, because linearization becomes an inaccurate method. This, of course, does not apply to methods that do not use linearization (Monte Carlo, for example, or Particle Swarm Optimization), on the other hand, these methods are much more complex and often less manageable.

2.2 Nonlinear Systems

For stable nonlinear models, one of the steps of the model calibration can be fitting the model steady-state responses towards the steady-state data. The steady-state data are often obtained by sampling the signals after sufficient time has elapsed from the last change of inputs; thus allowing the process to settle. This sampling approach is visualized in Figure 2.2.1. It assumes that the system is strictly stable and the effect of unmeasured disturbances is negligible. The steady-state calibration problem is easier to solve and it is easier to represent the global steady-state response by a limited data size which spans the whole operating range.

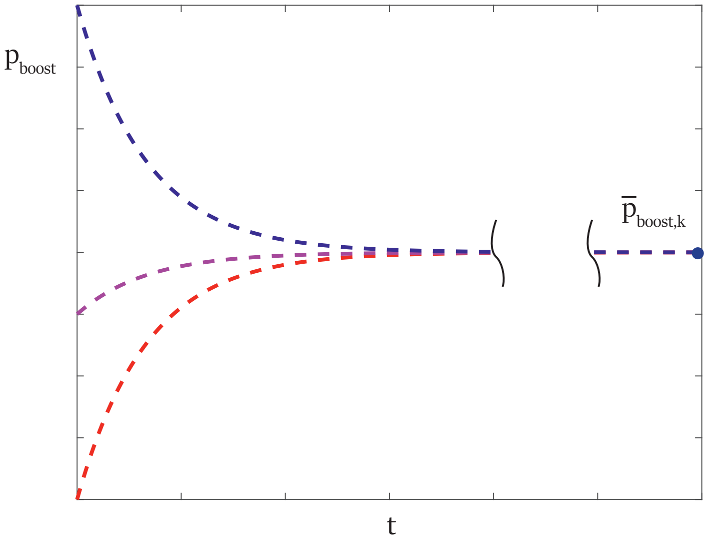

Figure 2.2.1: Process of steady state data collection.

Following text will concentrate on the steady-state calibration only, because the transient calibration is usually easier once the model steady-state is reasonable. In the steady-state, the model equations (1.1) become a set of equality constraints for each of the steady state responses fitted: $f(\mathbf{x}(k), \mathbf{u}(k), \mathbf{p}) = 0$.

2.3 System Calibration Methods

A model is designed to capture as much of the system behavior as possible. These differential equations often have parameters. Some parameters may be calculated from first principles or known from literature. However, it is extremely common that other parameters need to be fitted from observed data. In this section the most common calibration techniques are introduced. Just recall that the term calibration means to find the right/optimal parameters of the given system in its fixed structure.

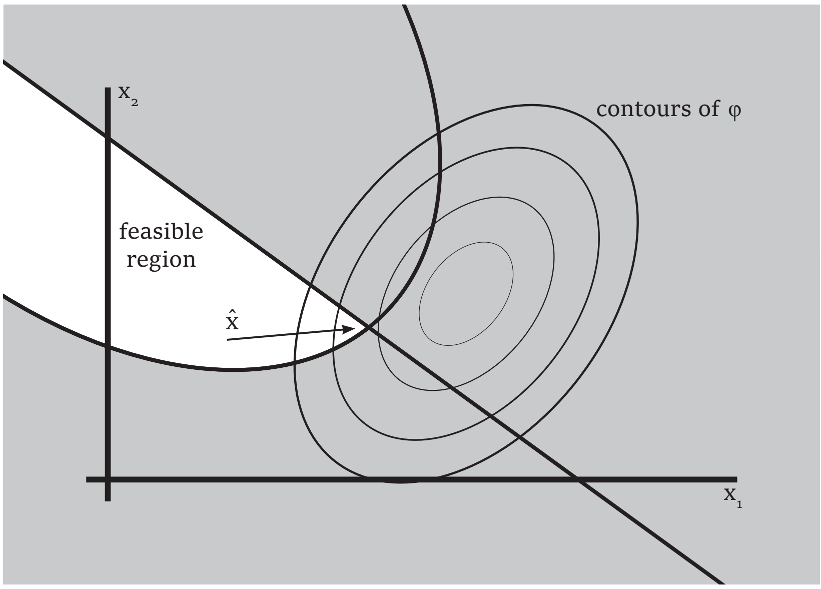

Figure 2.3.1: Geometrical representation of fundamental constrained optimization problem. For the sake of simplicity, there is no equality constraint.

A fundamental problem of mathematical optimization is to find an extreme (either maximum or minimum) of the objective function $\varphi(\mathbf{x})$

$$\begin{aligned} &\min_{\mathbf{x}\in R^n} \ \varphi(\mathbf{x}) \\ \text{subj. to } &q_j(\mathbf{x}) = 0, \quad j \in \mathcal{E} \\ &s_j(\mathbf{x}) \leq 0, \quad j \in \mathcal{I}, \end{aligned} \tag{2.1}$$

where $\mathcal{E}$ and $\mathcal{I}$ are sets of indicies for equality and inequality, respectively. Geometrical representation of the problem (2.1) is given in Figure 2.3.1.

2.3.1 Least-Squares Methods

Least-squares problems arise in many areas of applications, and may in fact be the largest source of unconstrained optimization problems [44]. The fundamental objective function used during solving least-square problems is

$$f(\mathbf{x}) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(\mathbf{x}) = \frac{1}{2}\|\mathbf{r}(\mathbf{x})\|_2^2, \tag{2.2}$$

where each r_j is a smooth function from $\mathbb{R}^n$ to $\mathbb{R}$. In the subsequent text will be assumed that $m \geq n$ for each residual r_j. Residuals are used to measure the discrepancy between the model and the observed behavior of the system. In practice this means that our model is fed by the same inputs as are measured on the system to obtain measured outputs $\mathbf{y}$, so $r_j = g_j - y_j$, where $\mathbf{g}$ is an output function of the model.

Evidently, optimal values of parameters can be found through finding a minimum of the objective function. To describe extremum search methods, the formula can be modified to nonlinear least squares, so that it is the sum of weighted squares of errors

$$J(\mathbf{p}) = \sum_{j=1}^{n} w_j \left(y_j - g(\mathbf{x}_j, \mathbf{p})\right)^2. \tag{2.3}$$

Here w_i is an element of weight vector $\mathbf{w}$. The objective function will be minimized due to vector of parameters $\mathbf{p}$. In general, vector of parameters $\mathbf{p}$ can be constrained by equality or inequality constraints

$$q_j(\mathbf{p}) = 0, \; j = 1, \ldots, n_1, \tag{2.4}$$

$$s_j(\mathbf{p}) \leq 0, \; j = 1, \ldots, n_2. \tag{2.5}$$

This is a fundamental problem of nonlinear programming (2.1) when the linear objective function is omitted. This problem will be more studied more in Chapter 5. Methods designed for finding minimum of these problems have mostly iterative character, which means that the sequence of parameters estimates $\mathbf{p}^{(0)}$, $\mathbf{p}^{(1)}$, $\ldots$, $\mathbf{p}^{(N)}$ is constructed. This sequence converges to optimal parameter values $\hat{\mathbf{p}}$ in successful cases.

In most methods of nonlinear programming there is a common procedure how the objective function can be minimized:

(S1) This is the first iteration ($i = 0$), so-called seed. First of all the initial condition, (initial estimate), $\mathbf{p}^{(0)}$ has to be chosen. This can be done, for example, based on approximate parameter estimation obtained from quasilinear regression.

(S2) Vector $\mathbf{v}^{(i)}$ has to be determined in the direction of i-th iteration step.

(S3) Value $\lambda^{(i)} > 0$ has to be chosen so that step $\lambda^{(i)}\mathbf{v}^{(i)}$ is acceptable, i.e. that next iteration

$$\mathbf{p}^{(i+1)} = \mathbf{p}^{(i)} + \lambda^{(i)}\mathbf{v}^{(i)} \tag{2.6}$$

minimizes the objective function

$$J(\mathbf{p}^{(i+1)}) < J(\mathbf{p}^{(i)}). \tag{2.7}$$

(S4) Terminal condition testing. If the "stopping criterion" is satisfied, then $\hat{\mathbf{p}} \sim \mathbf{p}^{(i+1)}$. If it is not satisfied, iteration index i is increased and algorithm continues to the step 2.

The choice of vector $\mathbf{v}^{(i)}$ and coefficient $\lambda^{(i)}$ determines individual methods of nonlinear programming. For the purpose of this thesis the methods using derivation of the objective function $J(\mathbf{p})$ will be described further.

2.3.2 Gauss-Newton Algorithm

Gauss-Newton method solves problem of minimization of the objective function which is in the form of nonlinear least-squares problem

$$\varphi(\mathbf{x}) = \frac{1}{2}\mathbf{h}^T(\mathbf{x})\mathbf{h}(\mathbf{x}) = \frac{1}{2}\sum_{j=1}^{N} h_j^2(\mathbf{x}), \tag{2.8}$$

where $\mathbf{h}(\mathbf{x})$ is a nonlinear function. To find a minimum of criterion function $\varphi(\mathbf{x})$ the first order condition must be satisfied

$$\nabla\varphi(\mathbf{x}) = \left(\frac{\partial\mathbf{h}(\mathbf{x})}{\partial\mathbf{x}}\right)^T \mathbf{h}(\mathbf{x}) = 0, \tag{2.9}$$

and the solution can be found using the Newton method

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \left(\frac{\partial}{\partial\mathbf{x}}\nabla\varphi(\mathbf{x})\right)^{-1} \nabla\varphi(\mathbf{x}). \tag{2.10}$$

For moderately-sized problems the Newton method typically converges much faster then gradient-descent methods, see [9]. The biggest disadvantage of the Newton method in this application is that it requires computation of the first and in particular the second derivatives of the objective function

$$\frac{\partial \varphi}{\partial \mathbf{x}} = \sum_{j=1}^{N} \left\{ \frac{\partial r_j}{\partial \mathbf{x}} r_j(\mathbf{x}) \right\}, \tag{2.11}$$

$$\frac{\partial^2 \varphi}{\partial \mathbf{x}^2} = \sum_{j=1}^{N} \left\{ \frac{\partial r_j^2}{\partial \mathbf{x}^2} r_j(\mathbf{x}) + \frac{\partial r_j}{\partial \mathbf{x}} \frac{\partial r_j}{\partial \mathbf{x}}^T \right\}. \tag{2.12}$$

This precise Hessian (2.12) is very often represented by neglecting the second-order derivative

$$\frac{\partial^2 \varphi}{\partial \mathbf{x}^2} = \sum_{j=1}^{N} \left\{ \frac{\partial r_j^2}{\partial \mathbf{x}^2} r_j(\mathbf{x}) + \frac{\partial r_j}{\partial \mathbf{x}} \frac{\partial r_j}{\partial \mathbf{x}}^T \right\} \approx \sum_{j=1}^{N} \left\{ \frac{\partial r_j}{\partial \mathbf{x}} \frac{\partial r_j}{\partial \mathbf{x}}^T \right\}. \tag{2.13}$$

The reason for neglecting second-order derivatives is:

- Functional values r_j are small in magnitude, the smallest near the minimum.
- The function is not too non-linear, so $\frac{\partial r_j^2}{\partial \mathbf{x}^2}$ are relatively small.

This neglect is also possible due to the linear approximation of the objective function around the current iteration.

Pure Gauss-Newton method is based on linearization of the objective function $\mathbf{h}(\mathbf{x})$ in point $\mathbf{x}^{(i)}$

$$\mathbf{h}(\mathbf{x}) \simeq \mathbf{h}(\mathbf{x}^{(i)}) + \nabla \mathbf{h}(\mathbf{x}^{(i)})(\mathbf{x} - \mathbf{x}^{(i)}). \tag{2.14}$$

New iteration is obtained with minimization of the norm of linear approximation. In next equation, the evaluation of the nonlinear function $\mathbf{h}(\mathbf{x}^{(i)})$ will be marked as $\mathbf{h}^{(i)}$ for clarity

$$\begin{aligned} \mathbf{x}^{(i+1)} &= \arg\min_{\mathbf{x} \in R^n} \frac{1}{2} \left[\mathbf{h}^{(i)} + \nabla \mathbf{h}^{(i)}(\mathbf{x} - \mathbf{x}^{(i)})\right]^T \left[\mathbf{h}^{(i)} + \nabla \mathbf{h}^{(i)}(\mathbf{x} - \mathbf{x}^{(i)})\right] = \\ &= \arg\min_{\mathbf{x} \in R^n} \frac{1}{2} \Big[\left(\mathbf{h}^{(i)}\right)^T \mathbf{h}^{(i)} + 2(\mathbf{x} - \mathbf{x}^{(i)})^T \left(\nabla \mathbf{h}^{(i)}\right)^T \mathbf{h}^{(i)} + \\ &+ (\mathbf{x} - \mathbf{x}^{(i)})^T \left(\nabla \mathbf{h}^{(i)}\right)^T \nabla \mathbf{h}^{(i)} (\mathbf{x} - \mathbf{x}^{(i)}) \Big]. \end{aligned} \tag{2.15}$$

After completing full square the minimal value of previous term will be

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \left(\nabla\mathbf{h}^T(\mathbf{x}^{(i)})\nabla\mathbf{h}(\mathbf{x}^{(i)})\right)^{-1}(\nabla\mathbf{h}(\mathbf{x}^{(i)}))^T\mathbf{h}(\mathbf{x}^{(i)}), \tag{2.16}$$

which is a pure Gauss-Newton method.

If the matrix $\left(\nabla\mathbf{h}^T(\mathbf{x}^{(i)})\nabla\mathbf{h}(\mathbf{x}^{(i)})\right)$ is singular, it has to be regularized with a choice of diagonal matrix $\Delta^{(i)}$ so that $\left(\nabla\mathbf{h}^T(\mathbf{x}^{(i)})\nabla\mathbf{h}(\mathbf{x}^{(i)}) + \Delta^{(i)}\right)$ is positive definite matrix. If the diagonal matrix $\Delta^{(i)}$ is chosen as $\Delta^{(i)} = a^{(i)}\mathbf{I}$, $(a^{(i)} > 0)$, the Levenberg-Marqardt method is obtained.

2.3.3 Levenberg-Marquardt Algorithm

Improvement of the Gauss-Newton method was done by American statistician Kenneth Levenberg in 1944 and in 1963 the algorithm was rediscovered by American statistician Donald Marquardt. The Levenberg-Marquardt algorithm adaptively varies the parameter updates between the gradient descent update and the Gauss-Newton update, see [17]

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \left(\nabla\mathbf{h}^T(\mathbf{x}^{(i)})\nabla\mathbf{h}(\mathbf{x}^{(i)}) + a^{(i)}\mathbf{I}\right)^{-1}(\nabla\mathbf{h}(\mathbf{x}^{(i)}))^T\mathbf{h}(\mathbf{x}^{(i)}). \tag{2.17}$$

If the coefficient $a^{(i)} \to 0$, the method is close to the Gauss-Newton method. If the coefficient $a^{(i)} \to \infty$, the method is close to steepest descend method. Now the problem is reduced to the choice of the coefficient $a^{(i)}$.

2.4 Shooting Methods

Shooting methods are coherent set of numerical methods that were originally developed as a numerical methods for solving boundary value problems in ordinary differential equations. These methods can be also used for system or steady-state calibration.

2.4.1 Single Shooting Methods

Single Shooting Method (SS) has its origins deep in the history of mathematics. Its name is derived from an analogy with a cannon shooting, where it is difficult to predict the impact of the target, so it must be done by observing the point of impact of the cannon ball in order to correct in the repetition of the shot.

Let $\boldsymbol{\psi}$ is given function and a is a given vector. Then the classical boundary value problem is defined as follows

$$\begin{aligned} \dot{\mathbf{x}}(t) = \boldsymbol{\psi}(t, \mathbf{x}(t)), \qquad & t \in (t_0, t_N), \\ x_j(t_N) \quad = & \quad a_j, \end{aligned} \tag{2.18}$$

where number of boundary values a_j can be less then a number of differential equations.

During solving, the SS method iteratively changes the initial condition $\mathbf{x}^{(i)}(t_0)$ with which the equation is solved numerically on a given time interval, thereby obtaining the solution $\mathbf{x}^{(i)}(t_N)$, which makes it possible to calculate how far the current solution is from given goal $\delta x_j^{(i)}(t) = x_j^{(i)}(t_N) - a_j$.

Using the variation of the differential equation (2.18), it is possible to calculate the sensitivity of changing the endpoint to the change of initial condition

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \boldsymbol{\psi}(t, \mathbf{x}(t)), \\ \dot{\mathbf{x}}(t) + \delta\dot{\mathbf{x}}(t) &= \boldsymbol{\psi}(t, \mathbf{x}(t) + \delta\mathbf{x}(t)), \\ \dot{\mathbf{x}}(t) + \delta\dot{\mathbf{x}}(t) &= \boldsymbol{\psi}(t, \mathbf{x}(t)) + \frac{\partial\boldsymbol{\psi}(t, \mathbf{x}(t))}{\partial\mathbf{x}(t)}\delta\mathbf{x}(t), \\ \delta\dot{\mathbf{x}}(t) &= \frac{\partial\boldsymbol{\psi}(t, \mathbf{x}(t))}{\partial\mathbf{x}(t)}\delta\mathbf{x}(t), \end{aligned} \tag{2.19}$$

and using this variation, the initial condition is set for next iteration. When linear differential equations are solved using SS method, the correct solution is found in one step.

It turns out, however, that the problem defined in this way is poorly conditioned. For some differential equations, it is difficult to solve it because the relationship between the state $\mathbf{x}(t_N)$ and the state $\mathbf{x}(t_0)$ can be very complex, especially if the

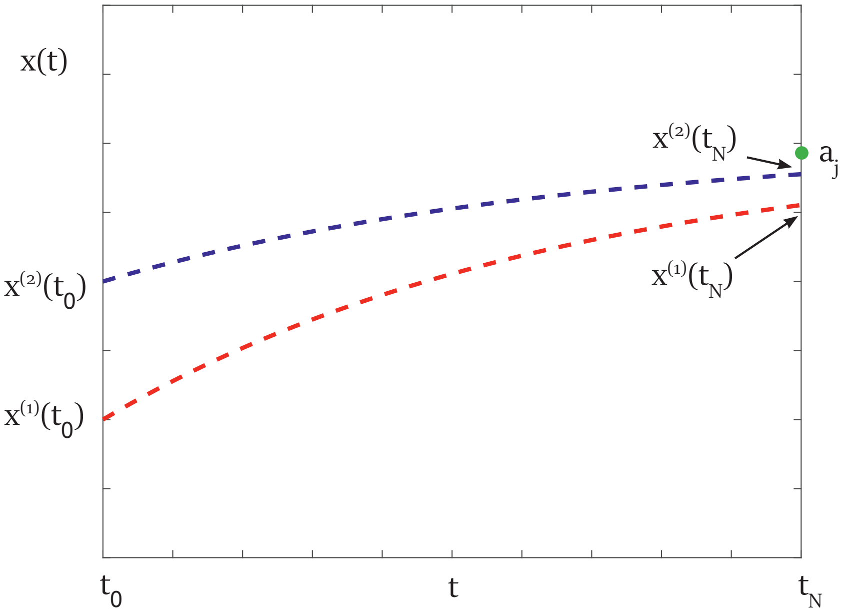

Figure 2.4.1: Typical run of the single-shooting algorithm.

given differential equation is nonlinear or the solution is required on a too long time interval. The convergence of the solution is also determined by how close the desired end condition is to the solution from the given initial condition. Typical run of the SS algorithm can be seen in Figure 2.4.1.

The task of calibrating parameters using the single shooting method is slightly different from the original problem definition. There is no requirement to hit the selected endpoint, but the requirement is to find a trajectory that minimizes the functional – to vary indefinite parameters of the differential equation in order to get the trajectory closer to the measured trajectory.

A similar situation occurs when we want to calibrate the system from steady-state data. The model is solved over a long time horizon to ensure that it achieved the steady-state, and the criterion of the method's success is in minimization difference between the solved steady state and the measurement, so $g(\mathbf{x}(t_N)) - \hat{\mathbf{y}} = 0$. Strictly speaking, this is no longer a single shooting problem as defined above.

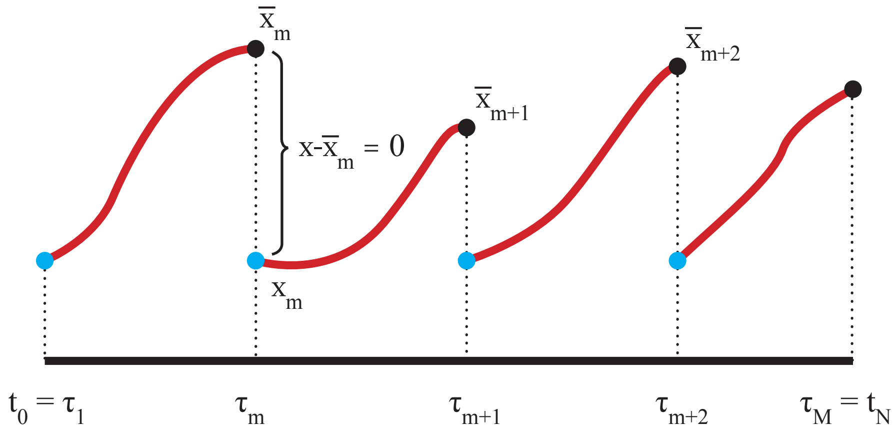

Figure 2.4.2: Graphical interpretation of the multiple-shooting method. Figure was inspired by [29].

2.4.2 Multiple Shooting Methods

Multiple shooting (MS) methods were introduced in [10]. These methods solve the common drawbacks of single-shooting methods by dividing the time-based solution into several intervals where the solution of the optimization problem is found separately. At the same time, equality constraints are added to the intervals end points, so the continuity of the trajectory across the solution horizon (t_0, t_N) is required. This situation is illustrated in the Figure 2.4.2.

Due to the division of the trajectory into several subintervals, the numerical stability of the whole procedure is improved. This division essentially divides (distributes) the original problem into several sub-problems that solve the original problem within their limited intervals.

2.5 Distributed Optimization Techniques

Distributed optimization is itself a very broad term. At the beginning, it can be said half-way that a complicated optimization problem can be divided into simpler tasks using the decomposition methods (as in the case of Multiple Shooting methods), and then solved separately. However, this division can not be accidental and must be supplemented by a number of equality constraints that maintain the integrity of the original task. Generally speaking, if the distribution of the original problem is well-designed, distributed problem solution is generally simpler and has better convergence to the optimum than the solution of the original, centralized task.

For a full understanding of the decomposition techniques, the basic terms and methods of decomposition as a Separable Problem, Complicating Variables, Primal and Dual decomposition, and after that the distributed identification algorithm work flow will be introduced. The following text is based on prof. Stephen Boyd's lessons, for more see [12] and [11].

In this section, symbols x, y and functions f, g have all different meaning than have been previously used.

2.5.1 Separable Problem and Complicating Variables

A simple optimization problem

$$\begin{aligned} \min_{x_1,x_2} \quad & \{f_1(x_1) + f_2(x_2)\}, \\ \text{subj. to} \quad & x_1 \in C_1,\ x_2 \in C_2, \end{aligned} \tag{2.20}$$

can be solved for x_1 and x_2 separately, either sequentially or in parallel, because solutions to these two problems are not tied together. Variables x_1 and x_2 are called private or local variables.

But if the problem is changed a little

$$\min_{x_1,x_2,y} \quad \{f_1(x_1,y) + f_2(x_2,y)\}, \tag{2.21}$$

we get the problem coupled together just by using a variable y. The variable y is here called complicating or coupling variable. In essence, this simple problem is basically the same as the problem solved by the distributed identification algorithm presented in this thesis.

The distribution of our task lies in the solution of many steady-state points in which the real system was measured. These so-called operating points form separable problems and can be solved separately with the given global vector of parameters.

A way to solve this problem is generally by a distributed iterative algorithm. Firstly, the complicating variable y is fixed, then the separable problems $x_1, \ldots, x_n$ are solved according to a given criterion. Based on how well the solution suits or fails in sum, the global identifier decides how to update the complicating variable for the next iteration.

2.5.2 Primal and Dual Decomposition

Two fundamental techniques of decomposition of this problem show, how to properly fix and work with the complicating variable from the example (2.21). These are the so called Primal and Dual decompositions.

Primal Decomposition

For a fixed complicating variable y the original problem (2.21) can be decomposed into two subproblems

$$\begin{aligned} \textit{subproblem}\ 1: &\quad \min_{x_1} f_1(x_1, y), \\ \textit{subproblem}\ 2: &\quad \min_{x_2} f_2(x_2, y), \end{aligned} \tag{2.22}$$

where optimal values of these subproblems are $\phi_1(y)$ and $\phi_2(y)$. Then the original problem (2.21) is equivalent to the problem

$$\min_y \quad \left\{ \phi_1(y) + \phi_2(y) \right\}, \tag{2.23}$$

which is called the master problem. The solution to this problem has already been mentioned in this section – for the fixed complicating variable y, in general the subproblems $1, \ldots, n$ will be solved, while the subproblems must also tell the optimizer of the complicating variable in some way how the selected y value matches them. For example, when using to solve a problem a gradient algorithm, the gradient $g_1 \in \nabla\phi_1(y), \ldots, g_n \in \nabla\phi_n(y)$ can be computed and after all subproblems are solved, the complicating variable can be updated in the direction of the sum of the gradients $y := y - a_k \sum_j g_j$ at the selected step of the a_k method. Graphical representation of Primal decomposition can be seen in Figure 2.5.1. Primal decomposition is called Primal because there the original, primal variable is distributed.

The primary benefit of Primal decomposition is working directly with a global variable. If the run of the distributed algorithm is interrupted at any time, it is always possible to determine the value of the global variable, which is also used in the task convergence monitoring. A major disadvantage of this decomposition is the case where constraints are attached to the problem that binds both the local and the global variables when the distribution of these constraints may not be easy, i.e.

$$\begin{aligned} s_1(x_1, y) &\leq 0, \\ &\vdots \\ s_n(x_n, y) &\leq 0. \end{aligned} \tag{2.24}$$

Here, different constraints are used in various sub-problems. It may happen that the solver, solving the original problem, sends within the i-th iteration y, such that it will not meet all the (2.24). This problem does not arise in Dual decomposition.

Dual Decomposition

In the original problem (2.21) the complicating variable y is now split into two new variables y_1 and y_2, and the problem is rewritten as:

$$\begin{aligned} \min_{x_1, x_2, y_1, y_2,} \quad & \{f_1(x_1, y_1) + f_2(x_2, y_2)\}, \\ \text{subj. to} \quad & y_1 = y_2, \end{aligned} \tag{2.25}$$

where y_1 and y_2 are local versions of complicating variable y and $y_1 = y_2$ is a consensus or consistency constraint.

As can be seen, two separable problems are created, which can be solved separately. But it must not be forgotten that the equality constraints must be incorporated into these subproblems. The lagrangian of this problem is

$$L(x_1, y_1, x_2, y_2) = f_1(x_1, y_1) + f_2(x_2, y_2) + \lambda^T(y_1 - y_2), \tag{2.26}$$

which is separable and these subproblems can be solved separately. λ is a Lagrange multiplier, and it has the meaning of price for violating the constrain $y_1 - y_2 = 0$. Therefore, the individual subproblems are

$$\begin{aligned} \text{subproblem 1 :} \quad & g_1(\lambda) = \min_{x_1, y_1} \left\{ f_1(x_1, y_1) + \lambda^T y_1 \right\}, \\ \text{subproblem 2 :} \quad & g_2(\lambda) = \min_{x_2, y_2} \left\{ f_2(x_2, y_2) - \lambda^T y_2 \right\}, \end{aligned} \tag{2.27}$$

with the master problem

$$g(\lambda) = \max_{\lambda} \left\{ g_1(\lambda) + g_2(\lambda) \right\}. \tag{2.28}$$

The solution to this problem proceeds in a similar way as in the case of Primal decomposition. The optimizer optimizing the λ determines the price for violating a constraint, and the subproblems $k = 1, \ldots, K$ are then solved. Then the price for a violation of the constraint is changed based on the requirements of individual subproblems, for example $\lambda := \lambda - a_k(y_2 - y_1)$, and the whole algorithm runs again. At the beginning of algorithm run, λ is usually low and increases during the iteration progresses. Gradually, the local variables $y_1, \ldots, y_n$ approach an optimal value. The advantage of this approach is that at any chosen price λ solution, subproblems always have solutions. Graphical representation of the Dual decomposition can be seen in Figure 2.5.1.

A major disadvantage of Dual decomposition is the fact, that during the run of the algorithm we do not have a value of original complicating variable, because each subproblem can send different value of $y_1, \ldots, y_n$ within the optimal solution of its subproblem. On the other hand, the great advantage of Dual decomposition is the fact, that if the original problem contains a constraint that combines local variables with complicating variables, these constraints can be pushed to the subproblem.

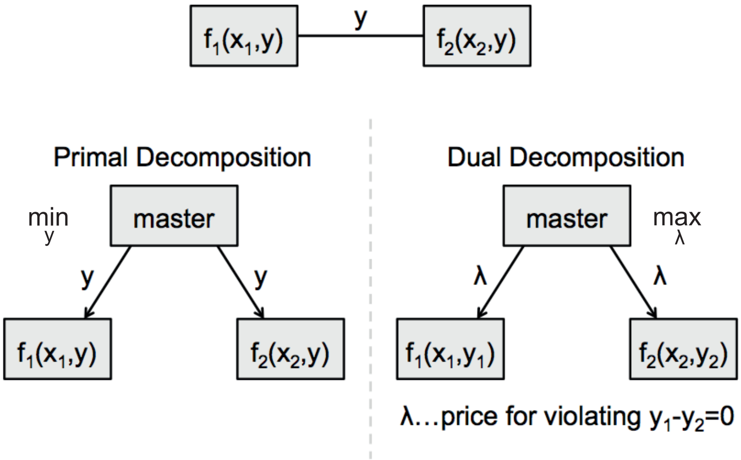

Figure 2.5.1: Graphical representation of Primal and Dual decompositions.

2.6 Distributed Constrained Least Squares Algorithm

Let us introduce the Quadratic Programming (QP) problem:

$$\begin{aligned} \min_{x_1,\dots,x_K,p} \sum_k \left\{ \frac{1}{2} \begin{bmatrix} x_k \\ p \end{bmatrix}^T Q_k \begin{bmatrix} x_k \\ p \end{bmatrix} + f_k^T \begin{bmatrix} x_k \\ p \end{bmatrix} \right\}, \\ \text{subj. to} \quad Ax_k \leq b, \\ Cp \leq d, \end{aligned} \tag{2.29}$$

where p is a global vector of a whole model parameters (complicating variable), x_k is a vector of local variables related to the specific subproblem. This vector is of the same type in each subproblem, i.e. it contains the same variables but they have different values in different subproblems.

Now we will take a small step aside and will try to explain the distribution of the calibration task, which we will deal with later on. The calibration of the system, the effort to find the right system parameters in a fixed system structure, is performed based on the measured data of the real system, since these data reflect the overall system behavior when they are measured well. Simply because the data contain as much information as possible about the behavior of the real system, the system is measured in various combinations of its inputs. If all the transient phenomena are dropped, the measured system is brought to a steady state, and the measured values then represent one steady-state point. If such measurements are repeated and the system inputs are changed methodically, we receive a set of measurements that indicate the behavior of the system in steady state. Since the measurements were made on the same system, it can be assumed that the parameters of this system were not changed.

While real-system measurement attempts to retrieve data rich in system behavior, the system calibration process proceeds in the "opposite" way of trying to retrieve that information from the data and determine the system's parameters in its structure.

The situation now resembles the problem defined above (2.29). As in the calibration task, one vector of global parameters (such as compressor map parameters or cross-section of a fully open valve) is the same for all subproblems, local variables (i.e. pressure between two components or gas temperature applied to the component) will vary in different subproblems.

Solving this problem in its original form is very demanding, if not unreal, due to a large number of variables. Just for the idea, if we had 10 global parameters and 100 operating points, where we need to optimize 5 local variables, the original optimization problem would have $10 + 5 \cdot 100 = 510$ variables. In addition, the Q and A matrices representing quadratic form and constraints would be very sparse.

The defined task (2.29) invites the use of the decomposition method and the division of the complicated task into a greater number of less demanding ones. For the algorithm described in this thesis, the method of Primal decomposition was chosen. The term global optimizer refers to an algorithm solving a top-level parameter optimization task, while local optimizer is an algorithm that solves individual subproblems.

For the fixed vector of parameters p satisfying constraints $Cp \leq d$, there are subproblems that need to be solved at each steady-state point, so for $\forall\, k = 1, \ldots, K$

$$\min_{x_k} \left\{ \frac{1}{2} \begin{bmatrix} x_k \\ p \end{bmatrix}^T Q_k \begin{bmatrix} x_k \\ p \end{bmatrix} + f_k^T \begin{bmatrix} x_k \\ p \end{bmatrix} \right\}, \qquad (2.30)$$
$$\text{subj. to} \quad Ax_k \leq b.$$

Solution of the Unconstrained Problem

For simplicity, we will now deal with a situation where the problem is distributed, but there are no constraints. Subproblem (2.30) will be

$$\min_{x_k} \left\{ \frac{1}{2} \begin{bmatrix} x_k \\ p \end{bmatrix}^T \begin{bmatrix} Q_{k,1} & Q_{k,2} \\ Q_{k,2}^T & Q_{k,3} \end{bmatrix} \begin{bmatrix} x_k \\ p \end{bmatrix} + \begin{bmatrix} f_{k,1} \\ f_{k,2} \end{bmatrix}^T \begin{bmatrix} x_k \\ p \end{bmatrix} \right\}, \qquad (2.31)$$

where the matrix Q_k was divided into the corresponding matrices $Q_{k,1}$, $Q_{k,2}$, $Q_{k,3}$, and the vector f_k was divided into the corresponding parts $f_{k,1}, f_{k,2}$. This quadratic form can be further modified to

$$J(x_k) = \frac{1}{2} \left(x_k^T Q_{k,1} x_k + 2\, p^T Q_{k,2}^T x_k + p^T Q_{k,3} p \right) + f_{k,1}^T x_k + f_{k,2}^T p. \qquad (2.32)$$

Since this is an unconstrained problem, the minimum of this form can be found analytically. The minimum of this criterion can be found from the zero gradient necessary condition with respect to x_k

$$\frac{\partial J(x_k)}{\partial x_k} = 0. \qquad (2.33)$$

For (2.32) the condition (2.33) will be

$$\frac{1}{2}(x_k^T(Q_{k,1} + Q_{k,1}^T) + 2\, p^T Q_{k,2}^T) + f_{k,1}^T = 0. \qquad (2.34)$$

This equation can be simplified (matrices $Q_{k,1}$ and $Q_{k,2}$ are symmetrical) and after equation transposition the optimal value of x_k^* is

$$x_k^* = -Q_{k,1}^{-1}(Q_{k,2} p + f_{k,1}). \qquad (2.35)$$

Now, the optimal value x_k^T can be substituted into the quadratic form (2.32). For this subproblem, a quadratic form depending only on the parameter vector will be found p

$$J_k(p) = \frac{1}{2}p^T(Q_{k,3} - Q_{k,2}^T Q_{k,1}^{-1} Q_{k,2})p + \left(f_{k,2}^T - f_{k,1}^T Q_{k,1}^{-1} Q_{k,2}\right) p - \frac{1}{2} f_{k,1}^T Q_{k,1}^{-1} f_{k,1}. \tag{2.36}$$

After subproblems are solved, an optimal values x_k^* are found, this quadratic form, which is dependent only on the global parameters, is returned to the global optimizer, i.e.

$$\begin{aligned} H_k(p) &= Q_{k,3} - Q_{k,2}^T Q_{k,1}^{-1} Q_{k,2}, & (2.37) \\ r_k^T(p) &= f_{k,2} - Q_{k,2} Q_{k,1}^{-1} f_{k,1}. & (2.38) \end{aligned}$$

Then the global optimizer solves a master problem according to the Primal decomposition rules (2.23)

$$\begin{aligned} \min_p & \left\{ \frac{1}{2} p^T \left(\sum_k H_k \right) p \;+\; \left(\sum_k r_k^T \right) p \right\}, \\ & \text{subj. to} \quad Cp \;\leq\; d, \end{aligned} \tag{2.39}$$

and will find an optimal parameter values p^*. So for quadratic function minimization without constraints the algorithm stops with an optimal value of p^* after first run. But as has been said, the constraints complicate the execution of the algorithm.

Constraints Management

After the subproblems are solved, so local optimizers have found optimal values x_k^*, the local optimizer calculates the fit of the measured values in the steady-state point and calculates how this fit would change if the parameters were changed. Otherwise, the result is a linear function how the local variables depend on the parameters ($x_k^* = \xi p + \vartheta$), which is valid within the constraints. This linear dependence is then fitted onto a local problem and the result will be the obtaining of the matrices $H_k(p),\ r_k^T(p)$ which are dependent only on the global parameters. Along with that, however, the local optimizer must also send constraints on vector of parameters, which says at which interval this approximation is valid $(\Gamma_k p \leq \delta_k)$.

For simplicity now, let suppose that there are only two linear constraints on x, $0 \leq x \leq 1$. This situation can be seen in Figure 2.6.1.

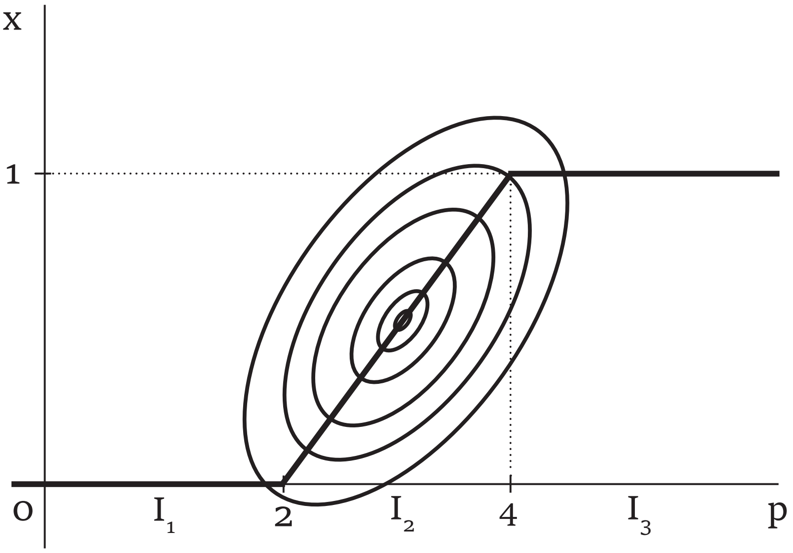

Figure 2.6.1: Graphical interpretation of the linear dependence of parameters on a local variable.

The dependence $x^*(p)$ is displayed in Figure 2.6.1 and is divided into three intervals $(I_1,\ I_2,\ I_3)$ as a result of the linear constraints $0 \leq x \leq 1$. At each of these intervals, another value x^* is optimal and therefore another linear dependence $x^*(p)$ is given. See the 2.6.1 table for details. However, at each interval there is another active constraint that is sent to the global optimizer along with the linear dependence. Thus, the local optimizer tells the global, that if this constraint is removed in the context of changing parameters, the local cost function will improve.

Interval	Optimal value of x	Linear dependence on p	Active set
I_1	$x^* = 0$	$x^*(p) = 0$	$x \geq 0 \Leftrightarrow p \leq 2$
I_2	$x^*(p)$	$x^*(p) = \xi p + \vartheta$	–
I_3	$x^* = 1$	$x^*(p) = 1$	$x \leq 1 \Leftrightarrow p \geq 4$

Table 2.6.1: Description of the linear dependence related to the Figure 2.6.1.

After receiving the information $H_k(p),\ r_k^T(p),\ \Gamma_k p \leq \delta_k$, global optimizer builds its master problem according to (2.23), and adds constraints $\Gamma_k p \leq \delta_k$ to its constraints to the vector of parameters

$$\min_{p} \left\{ \frac{1}{2} p^T \left(\sum_k H_k \right) p \ + \ \left(\sum_k r_k^T \right) p \right\}, \tag{2.40}$$
$$\text{subj. to} \quad \begin{bmatrix} C \\ \Gamma_1 \\ \vdots \\ \Gamma_K \end{bmatrix} p \ \leq \ \begin{bmatrix} d \\ \delta_1 \\ \vdots \\ \delta_K \end{bmatrix}.$$

If any of the constraints $\Gamma_k p \leq \delta_k$ is active, i.e. the constraint is limiting the improvement of the master problem cost function, the local optimizer, that added this constraint, has been requested within the new parameter values to recalculate its local problem and provide new matrices $H_k(p),\ r_k^T(p)$. The advantage of this approach is that in next iterations, only some local optimizers are requested to recalculate local problems (owners of constraints $\Gamma_k p \leq \delta_k$ that hinder the improvement of the cost function) to unblock future cost function improvement. If the set of active constraints is empty, the global optimizer resolves the master problem and finds the optimal value of the parameters p^*.

Physics is like sex: sure, it may give some practical results, but that's not why we do it.

Richard Feynman

3 Combustion Engine Modelling

Internal combustion engines (ICE) represent one of the most important technological success stories in the last century. For applications in the automotive industry, both gasoline (spark ignition) and diesel (compression ignition) are commonly encountered types of ICE. Diesel engines are mainly characterized by the separation of the air path from the fuel and commonly combined with supercharging. This idea first appeared on aircraft engines that were able to bring the plane to higher altitudes as more oxygen could be brought to the combustion chamber. Gradually this principle found its rightful place in both the gasoline and diesel (turbocharging) combustion engines. The main benefit is in increasing the power to weight ratio of the engine. Further development led to introduction of other engine components. Cooler, used to increase the density of the compressed air further increased the power to weight ratio of the engine. The Exhaust Gas Recirculation (EGR, the primary NOx emission reduction mechanism) managed to reduce exhaust emissions significantly. Research in the area of advanced turbines and compressors continued. The basic radial turbine principle has been extended with controls mainly to avoid over-speeding which could lead to the turbocharger destruction. Later (probably inspired by the Francis water turbine) also the possibility of turning the turbine blades (VGT) has been added. VGT increases the turbine efficiency over the broader interval of mass flows, which means that the turbocharger is working effectively over a broader range of engine operating points compared to the waste gated approach. The VGT technology is mainly used in heavy duty applications due to higher production cost.

At present, research of diesel combustion engines is mainly focused on reducing emissions of pollution gases meet the emission standards (Euro 6+, VI+). This is achieved primarily by improving the EGR technology and the introduction of filters and catalysts to the exhaust flow. Diesel Particulate Filter – DPF, and Selective Catalytic Reduction – SCR, are the most common technologies. It is possible to meet today's stringent standards only when several of these technologies (usually EGR, DPF and SCR or LNT) are combined.

3.1 System Overview

In Figure 3.1.1, one common diesel engine structure can be seen. This engine, which is controlled by the air path, in the first approach consists of the turbocharger and of the engine, which is equipped with a cooler and EGR branch.

In Fig. 3.1.2 a structure of turbocharger can be seen. The ordinary turbocharger consists of a compressor and a turbine that are on a common shaft. The main task of the compressor is to create overpressure in front of the combustion chamber and due to this extra-air burn more fuel in the chamber and thereby increase the engine power at its constant volume, fuel mixture ratio and engine speed. This is the main principle of engine supercharging. In recent years, variable geometry compressors have also begun to emerge in the shape of variable geometry turbines as a solution to improve the compressor performance and stability, see [43].

The main task of the turbine is to deliver the power to the compressor from the exhaust gas energy. This partly uses the thermal energy of the exhaust gas, but on the other hand, the engine is forced to work in the load. In general, there are

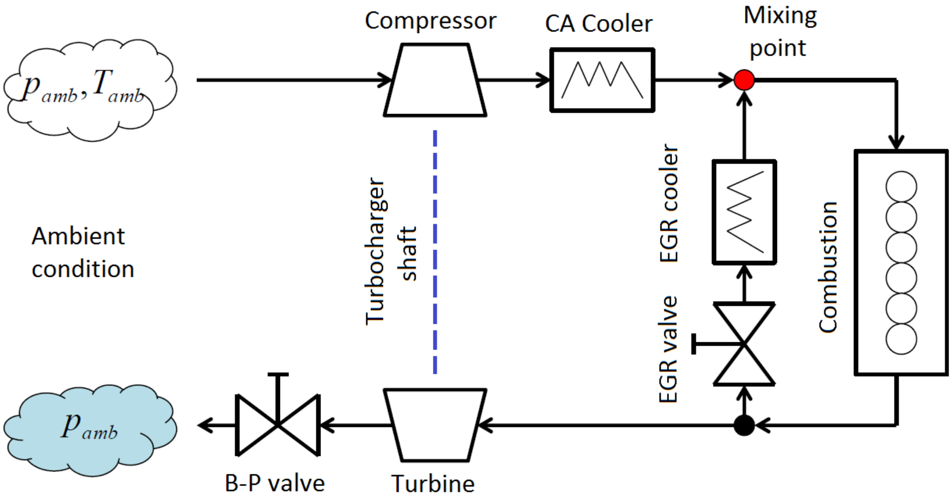

Figure 3.1.1: Typical structure of Diesel internal combustion engine.

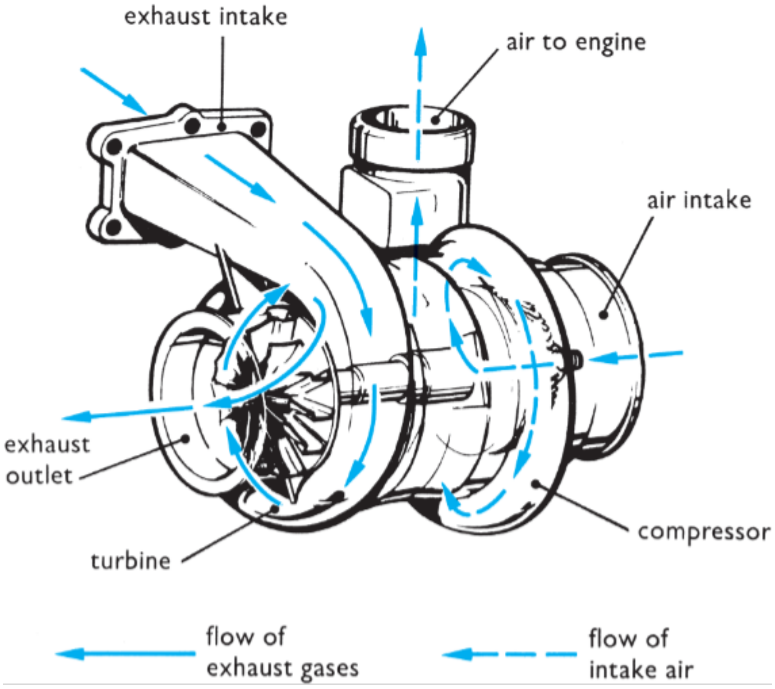

Figure 3.1.2: Structure of diesel turbocharger [3].

three common types of the turbine:

- *Simple radial turbine* – could be also called free of floating turbine, which are the simplest of all these turbines and were used in the first applications of turbocharging. In this turbine, the turbine blades are fixed and the turbine efficiency function is given by the pressure ratio across the turbine, which result in a strong "turbo-effect", so it is necessary first to develop an adequate pressure ratio to achieve a good efficiency of the turbocharger.

- *Waste-gate turbine* – it is a classic turbine equipped with a bypass valve. By means of this valve, a part of the hot exhaust gas from the engine can be displaced behind the turbine, thereby protecting the turbine against undesirable states due to the maximum filling pressure, but also to partially control the turbocharger speed. At present, this type of turbine is most used.

- *Variable-geometry Turbine (VGT)* – also known as Variable Nozzle Turbine (VNT). The variable geometry of the turbine blades is another upgrade of the simple radial turbine and is principally derived from the water inlet of the Francis water turbine. A switchgear is inserted into the turbine stator, which, due to the positional change of the blades, changes the winding supply and the angle at which the air is applied to the blades of the turbine. By changing this flow rate, the rate of incoming flue gases changes at constant pressure, which affects the turbine speed and thus the filling pressure in the intake manifold. Variable-geometry Turbine is currently mainly used in heavy-duty applications, primarily due to the high purchase price, which is also associated with the demand for turbine control.

It is obvious that due to a higher pressure the compressed air has a higher temperature. For this reason, a cooler is placed before the air entering the engine, see Fig. 3.1.3. In order to further support the idea of supercharging, the same cooler is used in the EGR branch.

EGR is used in all modern ICE's (both gasoline and diesel). A portion of the exhaust gas is fed back into the intake manifold and takes part in the combustion process again. Combustion thus involves not only the air and fuel, but also the additional three-atomic inert components (CO_2, H_2O). In the diesel engine up to 50 %

Figure 3.1.3: 4 stroke diesel engine intercooler [1].

of the exhaust gas is fed back of the cylinder volume and thus replacing the useful (power producing) air. Inert ingredient which is introduced to boost the overall heat capacity of the cylinder content thus reduces the peak combustion temperature and it eliminates the production of NO_x (NO_x is typically formed when the peak combustion temperature is above 2500 °C). The EGR branch typically consists of an EGR cooler and a valve.

Valves are used for two different purposes in engines – first, as was said, they control exhaust gas recirculation (EGR), and second, they ensure temporary pressure increase in the exhaust manifold, when the back pressure valve or waste-gate turbine is used. Alternatively, the internal throttle valve can decrease the intake manifold pressure temporarily. In both cases, the range of pressure difference values across the engine is extended. In the gasoline engines, the intake throttle is the main device to control the engine power controlling its air flow. However, the modern efficient gasoline engines are also using the valve timing for the same purpose, which is more energy efficient.

The intake throttle valve (ITV), see Fig. 3.1.4, and back pressure valve (BPV) have both the same purpose but they are placed at different locations in the engine. Typical BPV position can be seen on Figure 3.1.1, ITV is usually placed in the intake before air/EGR mixing point. Thanks to their operation, the pressure difference across the EGR valve can increase (decreasing pressure in intake manifold closing ITV or increasing the pressure in the exhaust manifold with the BPV),

Figure 3.1.4: Typical valve body [2].

which can be useful in situations when the EGR has not enough pressure difference to work effectively.

A heart of ICE is a combustion model, which in most cases is divided into several parts, especially the mass flow model and the combustion model based on heath-flow models. Of course, it is intended to describe the behavior of the combustion chamber processes, both in terms of mixing and combustion of fuel, and in terms of the power transferred to the shaft and the energy contained in the exhaust gases.

All introduced components will be described later in this Chapter in detail.

3.2 Common Modelling Principles

The modelling of internal combustion engines can be divided into three main directions, mainly according to the use of the resulting model.

3.2.1 Mean Value Models

One of the earliest papers proposing MVM for engine systems is [37]. Mean Value Models of ICE are based on physical first principles, which means that all relationships used in the models are derived from basic physical and thermodynamical principles. However, the gas flows are considered independent of the intake and exhaust gas opening, but equivalent quasi steady-state mean flows are assumed. This is sometimes described as approximating the engine with an equivalent engine which captures the general relationships except of the fast phenomena. Using these principles, the whole model structure can be derived from models of few components which are then connected appropriately.

All these models are Control Oriented Models (COM). This means that they have a great popularity in subsequent engine control using standard controller techniques, such as PID – Proportional-Integral-Derivative or advanced control techniques – the nowadays very popular predictive regulators – MPC – Model Predictive Control and its variations. As will be shown below, MVM mostly consist of a set of algebraic equations supplemented by nonlinear differential equations describing the main engine dynamics.

3.2.2 Another Types of Models Used to Represent ICE

Of course, MVM is not the only modelling technique used to model ICE. However, because this thesis does not deal with this issue, only a very rough summary will be given and the reader may find more details in the enclosed literature.

1D Models

These kinds of models are typically based on strong physical background, which leads to a large set of ordinary differential equations which describe the whole engine dynamics in one spatial dimension and in the time domain. The models range from a simple model requiring only thermodynamic properties to a complex model demanding full combustion kinetics, transport properties, poppet valves, flow characteristics etc., see [42].

2D or 3D Models

The model is characterized by non-linear differential equations or partial-differential equations in two or three spatial dimensions and in the time domain. It is based on a strong physical background of fluid dynamics combined with complex combustion chemistry of the engine. Complex geometry of the whole engine with boundary conditions is modelled too. The result of the modelling process is usually full two/three dimensional Computational Fluid Dynamics (CFD) model, which is capable of solving general transport equations for fluid flow, heat transfer and equations of combustion chemistry, see [39], [36] and [34]. As an application of CFD models, there could be for example models of Intake systems, In-cylinder flow and combustion, Engine block, Exhaust and after-treatment system, Fuel supply system, Engine ancillaries and so on.

3.3 Mean Value Model of Internal Combustion Engine

Following the subsection (3.2.1), the physics of the main engine components will be discussed with emphasis on the validity ranges of those models. Most of the models are valid under specific conditions only. The description of the symbols used in the following can be found in abbreviations.

3.3.1 Valves

Simple Valve Model

Valve equation can be derived from Bernoulli's law (in this case, the flow is incompressible) and from ideal gas law equation, see [18].

Inputs: p_{in} [kPa], p_{out} [kPa], T_{in} [K], u [-]
Outputs: $\dot{m}$ [kg/s]

$$\begin{aligned} \rho &= \frac{p_{\mathrm{in}}}{R \cdot T_{\mathrm{in}}}, \\ \dot{m} &= c_{\mathrm{d}} \cdot A(u) \cdot \sqrt{2\rho} \cdot \sqrt{p_{\mathrm{in}} - p_{\mathrm{out}}}. \end{aligned} \tag{3.1}$$

There the function $A(u)$ means that the open valve cross section is a nonlinear function and depends on the control signal u. It is very important to note that in this formulation the passage of fluid in direction that $p_{\mathrm{in}} - p_{\mathrm{out}} > 0$ is modelled only.

Orifice Valve Model

Equation (3.1) can be found in literature as a quadratic restriction (due to a relationship between pressure difference and mass flow square) and it is a simplified version of the variable restriction which is a most general model of restriction ("orifice" equation)

Inputs: p_{in} [kPa], p_{out} [kPa], T_{in} [K], u [-]
Outputs: $\dot{m}$ [kg/s]

$$\begin{aligned} \dot{m} &= A(u) \cdot \frac{p_{\text{in}}}{\sqrt{R \cdot T_{\text{in}}}} \Psi\left(\Pi\right), \; \Pi = \frac{p_{\text{in}}}{p_{\text{out}}}, \\ \Psi(\Pi) &= \begin{cases} \sqrt{\frac{2\gamma}{\gamma-1}\left(\Pi^{\frac{2}{\gamma}} - \Pi^{\frac{\gamma+1}{\gamma}}\right)} & \text{if } \Pi \geq \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \\ \sqrt{\gamma\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{\gamma-1}}} & \text{else,} \end{cases} \end{aligned} \tag{3.2}$$

where the upper part of function $\Psi(\Pi)$ is related to the supersonic flow through the valve.

The orifice equation (3.2), or quadratic restriction (3.1) is used to modell all throttles and valves in ICE (EGR, ITV, BPV, and so on). The air filter and after-treatment flow restriction can be modeled by (3.1) with a constant open cross section A.

3.3.2 Heat Exchanger

Cooler is now an integral part of the turbocharging. The goal is to get the maximum mass of the fresh air, whose density increases with decreasing temperature, into the cylinder. Because the air temperature increases when air is being compressed according to the ideal gas law, the air is cooled before entering the cylinder, by passing the intercooler heat exchanger. Temperature of the compressed air at the outlet of the intercooler temperature reaches approximately that of the originally aspirated air. The EGR coolers are used for the same purpose: to cool the recirculated exhaust gas.

From this point of view the heat exchanger decreases the energy flow

Inputs: T_{in} [K]
Outputs: T_{out} [K]

$$T_{\text{out}} = T_{\text{in}} - \eta(T_{\text{in}} - T_{\text{ref}}). \tag{3.3}$$

Combining heat exchange equation (3.3) with quadratic flow restriction equation (3.1) for a constant open cross section area A, we get a complete model of the cooler.

More complex models of coolers are designed primarily using 1D and 3D modelling methods, see [15], [41].

3.3.3 Mixing Plenum

As will be seen below, mixing plenum is another important part of ICE. Here we try to model the interconnection of the intake duct with the EGR duct. From the first principles, the resultant flow is simply a sum of both flows and resultant temperature of the mixture can be calculated as a weighted average of both flows, where weights are constant pressure specific heats of the corresponding flows, see [18].

Inputs: $\dot{m}_1$ [kg/s], $\dot{m}_2$ [kg/s], T_1 [K], T_2 [K]
Outputs: $\dot{m}$ [kg/s], T_{out} [K]

$$\begin{aligned} \dot{m} &= \dot{m}_1 + \dot{m}_2, \\ T_{\text{out}} &= \frac{c_{p1}\dot{m}_1 T_1 + c_{p2}\dot{m}_2 T_2}{c_{p1}\dot{m}_1 + c_{p2}\dot{m}_2}. \end{aligned} \tag{3.4}$$

3.3.4 Turbocharging

Turbocharger modeling is typically one of the most problematic parts of mean-value internal combustion engine components modelling. These difficulties are mainly connected with compressor and turbine flow and efficiency expression due to the complex physic which has to be used. This situation is partially solved in Mean-Value models using empirical functions which try to fit the compressor map using parameters, see Figure 3.3.1. Empirical models are very helpful in this situation, mainly because of their simplicity, but also bring along a number of problems as will be seen later.

Compressor

The compressor model often produces severe nonlinearities in the **f** function of the turbocharged ICE model, see [26]. The compressor element is highly nonlinear although this nonlinearity is partly attenuated when it is plugged into the overall model. The reason for this is the negative feedback in the engine structure. Taking the compressor flow model introduced by [24], the expressions for the mass flow through the compressor will be:

Inputs: p_{in} [kPa], p_{out} [kPa], T_{in} [K], N [1/min]

Outputs: $\dot{m}$ [kg/s]

Parameters: X

$$\begin{aligned}
U_c &= \frac{\pi}{60}\sqrt{\frac{T_{\text{ref}}}{T_{\text{in}}}}dN, \\
\Psi &= \frac{2C_pT_{\text{in}}}{U_c^2}\left[\left(\frac{p_{\text{out}}}{p_{\text{in}}}\right)^{\frac{\gamma-1}{\gamma}}-1\right], \\
M &= \frac{U_c}{\gamma RT_{\text{in}}}, \\
\Phi &= \frac{(k_5+k_6M)\Psi-(k_1+k_2M)}{(k_3+k_4M)+\Psi}, \\
\dot{m} &= \frac{p_{\text{in}}}{p_{\text{ref}}}\sqrt{\frac{T_{\text{ref}}}{T_{\text{in}}}}\frac{p_{\text{in}}}{RT_{\text{in}}}\frac{\pi}{4}d^2\Phi U_c.
\end{aligned} \tag{3.5}$$

Another model of compressor can be seen in [8] and [30]. This model is based on thermodynamic energy transformation, but it was developed using regression analysis. Model of compressor mass flow and outlet temperature in this case is

Inputs: p_{in} [kPa], p_{out} [kPa], T_{in} [K], N [1/min]

Outputs: $\dot{m}$ [kg/s]

Parameters: $k_{1,2,3,4}$, $s_{1,2,3,4}$ [-]

$$\begin{aligned}
\dot{m} &= k_1\left(1-\frac{p_{\text{out}}}{p_{\text{in}}}\right)+Nk_2\sqrt{1-\frac{p_{\text{out}}}{p_{\text{in}}}}+Nk_3\sqrt[4]{1-\frac{p_{\text{out}}}{p_{\text{in}}}}+Nk_4, \\
T_{\text{out}} &= s_1\dot{m}^2+Ns_2\dot{m}+N^2s_3+s_4T_{\text{in}},
\end{aligned} \tag{3.6}$$

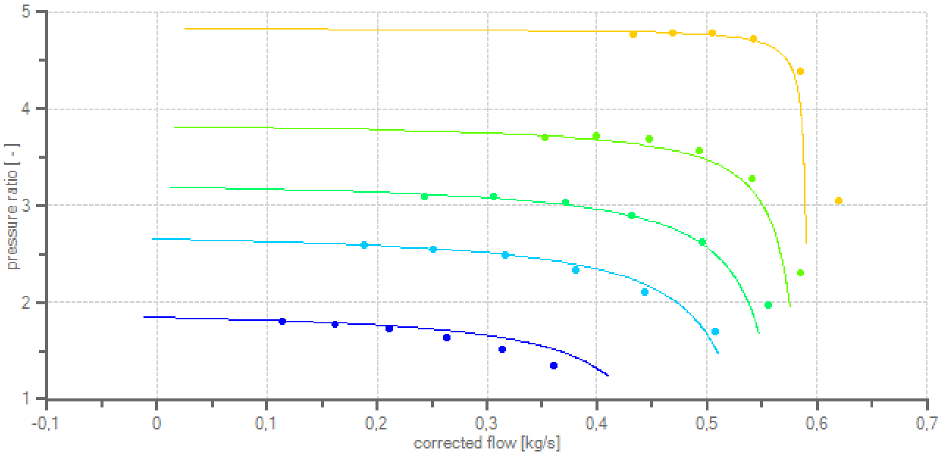

Figure 3.3.1: Typical parameters fitting from the compressor map.

where $k_{1,2,3,4}$ and $s_{1,2,3,4}$ are constants from regression analysis. The advantage of these two models is that they have same inputs and outputs.

Turbine

In the following the model of VGT turbine will be introduced which is the most complex model from the three model variations discussed above, and the model most used in the contemporary advanced ICE projects.

In Figure 3.3.2, the dependence of turbine mass flow on the expansion ratio (ratio of inlet to outlet pressures) is shown. According to that figure, the turbine mass flow $\dot{m}$ is also a weak function of turbine speed N. The mass flow dependency on the VGT position is usually determined by heuristic empirical functions (see [18])

Inputs: u [-], p_{in} [kPa], p_{out} [kPa], T_{in} [K], N [1/min]
Outputs: $\dot{m}$ [kg/s]

$$\dot{m} \quad = \quad \frac{p_{\text{in}}}{\sqrt{T_{\text{in}}}} \cdot c_t \cdot \sqrt{\left[1 - \left(\frac{p_{\text{in}}}{p_{\text{out}}}\right)_t^{k_t}\right]} \mathrm{d}N, \tag{3.7}$$

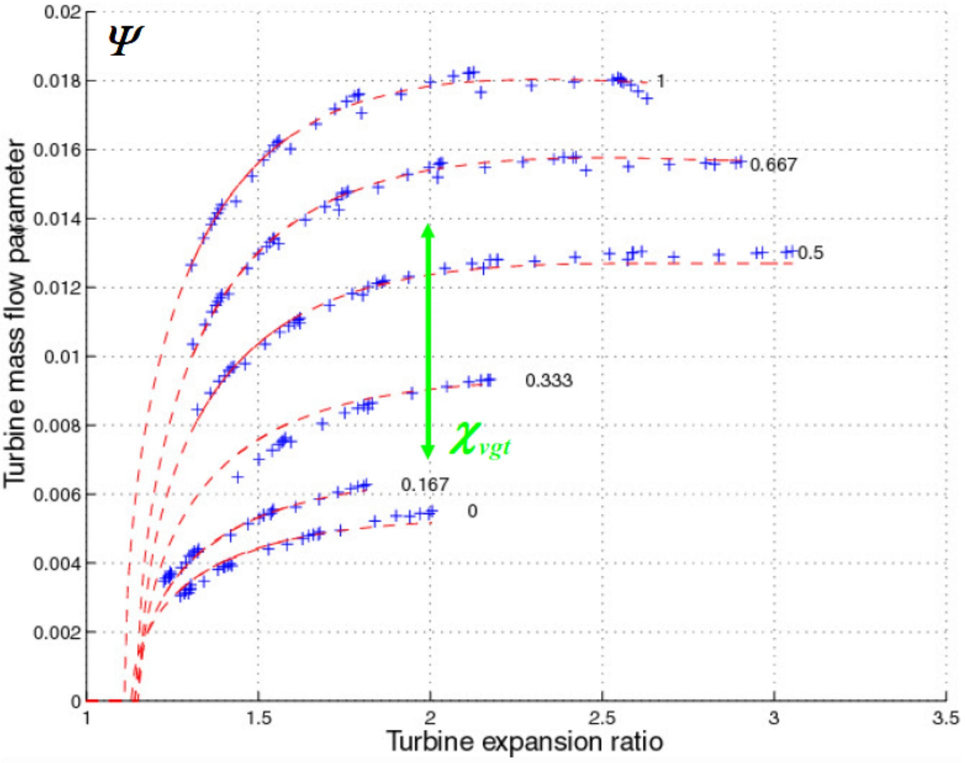

Figure 3.3.2: Fitting of the turbine model parameters. Figure was taken from lectures of prof. Ilja Kolmanovsky.

where parameters c_t and k_t can be determined with a least square algorithm.

Biteus-Muller Model

Another turbine flow model can be found in [8] and [30]. In the same way as in the compressor model (3.6), this model is based on thermodynamic energy transformation and was developed using the regression analysis

Inputs: p_{in} [kPa], p_{out} [kPa], T_{in} [K]

Outputs: $\dot{m}$ [kg/s]

$$\frac{p_{\text{in}}}{p_{\text{out}}} = t_1\left(\frac{\dot{m}\sqrt{T_{\text{in}}}}{p_{\text{out}}}\right)^2 + t_2\frac{\dot{m}\sqrt{T_{\text{in}}}}{p_{\text{out}}} + t_3, \tag{3.8}$$

$$\dot{m} = \frac{p_{\text{out}}}{\sqrt{2t_1 p_{\text{in}}}}\left[-t_2 + \sqrt{t_2^2 + 4t_1\left(\frac{p_{\text{in}}}{p_{\text{out}}} - t_3\right)}\right]. \tag{3.9}$$

From equation (3.8) the constants $t_{1,2,3}$ can be fitted using the least squares approach. These are used in turbine mass flow model (3.9). In this approach, turbine outlet temperature T_{out} is not modeled.

Similarly as the compressor power, the turbine power can be expressed using isotropic efficiency function $\eta_t(p_{\text{in}}/p_{\text{out}}, N/\sqrt{T_{\text{in}}}, p)$

$$P_t = \eta_t(p_{\text{in}}/p_{\text{out}}, N/\sqrt{T_{\text{in}}}, p)\dot{m}c_{\text{p}}T_{\text{in}}\left[1 - \left(\frac{p_{\text{in}}}{p_{\text{out}}}\right)^{-\frac{\gamma-1}{\gamma}}\right]. \tag{3.10}$$

Turbocharger Shaft

The model of a turbocharger shaft is derived from basic mechanics and reflects the energy disparity between the compressor and turbine. The mechanical behaviour of a combined turbocharger group can be described as follows:

Inputs: P_{c} [W], P_{t} [W], N [1/s]
Outputs: $\dot{N}$ [1/s^2]

$$M_{\text{c}} = P_{\text{c}}/N, \quad M_{\text{t}} = P_{\text{t}}/N, \tag{3.11}$$

$$\frac{\text{d}}{\text{d}t}N(t) = \frac{1}{J_{\text{tc}}}[M_{\text{t}}(t) - M_{\text{c}}(t) - M_{\text{f}}(t) + M_{\text{a}}(t)], \tag{3.12}$$

where J_{tc} is the turbocharger's rotational inertia and $M_i(t)$ represents the torques acting on the shaft. The turbine (produced) and compressor (absorbed) torques can be computed from isotropic power. The remaining two torques are additional friction losses and possible external auxiliary torques and in our application both are assumed as zero.

3.3.5 Combustion

The basic approach to combustion modelling is to divide the model into two parts – a model of the mass flows based on the volumetric pump model and the combustion model based on heath-flow model. Mass flow model describes the fact that during the intake phase of the cylinder cycle, air fills the cylinder. This filling depends on many factors, but the most important are engine speed, intake manifold pressure, and intake manifold temperature

Inputs: δ [kg/stroke], p_{in} [kPa], p_{out} [kPa], T_{in} [K], N_{eng} [rpm]
Outputs: $\dot{m}$ [kg/s]

$$\begin{aligned} \dot{m}_{\text{in}} &= \eta_{\text{vol}} \frac{V_{\text{d}} N_{\text{eng}} p_{\text{in}}}{2RT_{\text{in}}}, \\ \eta_{\text{vol}} &= f_{\eta_{\text{vol}}}(N_{\text{eng}}, T_{\text{in}}), \end{aligned} \tag{3.13}$$

where η_{vol} is volumetric efficiency which represents the ratio between the mass induced into the engine and mass ideally induced assuming the intake gas density and the engine displacement. This function needs to be mapped from the engine speed and the intake manifold temperature.

During the exhaust phase, the exhaust gases are pressed out of the cylinder. Depending on the engine valves timing, part of exhaust gases can be pressed into the intake manifold (this is called internal exhaust gas recirculation – IEGR), but most of the gases are pushed into exhaust manifold. In this case, the flow out of the engine equals the sum of the flow into the engine and the amount of fuel injected

$$\begin{aligned} \dot{m}_{\text{out}} &= \dot{m}_{\text{in}} + m_{\text{fuel}}, \\ m_{\text{fuel}} &= \delta n_{\text{cyl}} N_{\text{eng}}. \end{aligned} \tag{3.14}$$

Engine output temperature can be modelled in several ways. The simplest approach probably uses the ideal Otto cycle, which can be seen in [18]. In [8] the output temperature is described with nonlinear equation system

Inputs: p_{in} [kPa], p_{out} [kPa], m_{fuel} [kg], T_{in} [K]
Outputs: T_{out} [K], T_1 [K], q_{in} [J/kg], x_{r} [-]

$$\begin{aligned} T_{\text{out}} &= T_1 \left(\frac{p_{\text{out}}}{p_{\text{in}}}\right)^{\frac{\gamma-1}{\gamma}} \left(1 + \frac{q_{\text{in}}}{c_{\text{v}} T_1 r_{\text{c}}^{\gamma-1}}\right)^{\frac{1}{\gamma}}, \\ q_{\text{in}} &= \frac{m_{\text{fuel}}\, q_{\text{HV}}}{\dot{m}_{\text{in}} + m_{\text{fuel}}}(1 - x_{\text{r}}), \\ x_{\text{r}} &= \frac{1}{r_{\text{c}}} \left(\frac{p_{\text{out}}}{p_{\text{in}}}\right)^{\frac{1}{\gamma}} \left(1 + \frac{q_{\text{in}}}{c_{\text{v}} T_1 r_{\text{c}}^{\gamma-1}}\right)^{-\frac{1}{\gamma}}, \\ T_1 &= x_{\text{r}} T_{\text{out}} + T_{\text{in}}(1 - x_{\text{r}}), \end{aligned} \tag{3.15}$$

where the variables q_{in} and x_{r} represent specific energy of the charge per mass and residual gas fraction. This nonlinear system has to be solved in every combustion subsystem evaluation.

3.4 Input/Output System Review

For the purpose of steady-state calibration, there is no need for a dynamic system model, the one that describes the behavior of the system in steady-state suffices. Such a model can be seen in Fig. 3.4.1.

The presence of equations describing the system in steady-state instead of system dynamics is accompanied by the presence of variables, which need to be added to the steady-state model so that the model is complete, can be simulated and calibrated. This is due to the transition of dynamical differential equations to a steady-state form ($\dot{x} = f(x) \rightarrow 0 = f(x)$), thus also changing the explicit form of the equations to the implicit. The initial values of these variables are taken from the measurement data, and these variables are added to the optimization vector x_k within the calibration in individual steady-state points. These variables will be in the subsequent text referred to as independent variables.

Each such introduction of a free variable is also accompanied by the addition of an additional equality constraint that needs to be satisfied within the identification process. In the case of a diesel combustion engine, these additional equality constraints are the laws of flow and energy conservation.

In the case of the schema shown in Figure 3.4.1, these variables and corresponding equality constraints, used in the model, are:

Independent variable	Corresponding constraint
intake manifold pressure	$\dot{m}_{\text{eng}} - \dot{m}_{\text{IM}} - \dot{m}_{\text{fuel}} = 0$
exhaust manifold pressure	$\dot{m}_{\text{eng}} - \dot{m}_{\text{EGR}} - \dot{m}_t = 0$
exhaust manifold temperature	$T_3 - T_{\text{eng\_out}} = 0$
output turbine pressure	$\dot{m}_t - \dot{m}_{\text{BPV}} = 0$
turbocharger speed	$P_{\text{trb}} + P_{\text{comp}} = 0$

Table 3.4.1: Independent variables and the corresponding constraints contained in the Figure 3.4.1.

In this table, the intake manifold mass flow $\dot{m}_{\mathrm{IM}}$ is computed according to the mixing plenum flow equation (3.3.3), so $\dot{m}_{\mathrm{IM}} = \dot{m}_c + \dot{m}_{\mathrm{EGR}}$. Furthermore, while turbine delivers positive power, the compressor consumes power, so its power is negative.

Other variables that need to be connected with the model are control inputs such as EGR valve position (u_{EGR}), engine speed (N_{eng}), injection quantity (u_{IQ}), etc.

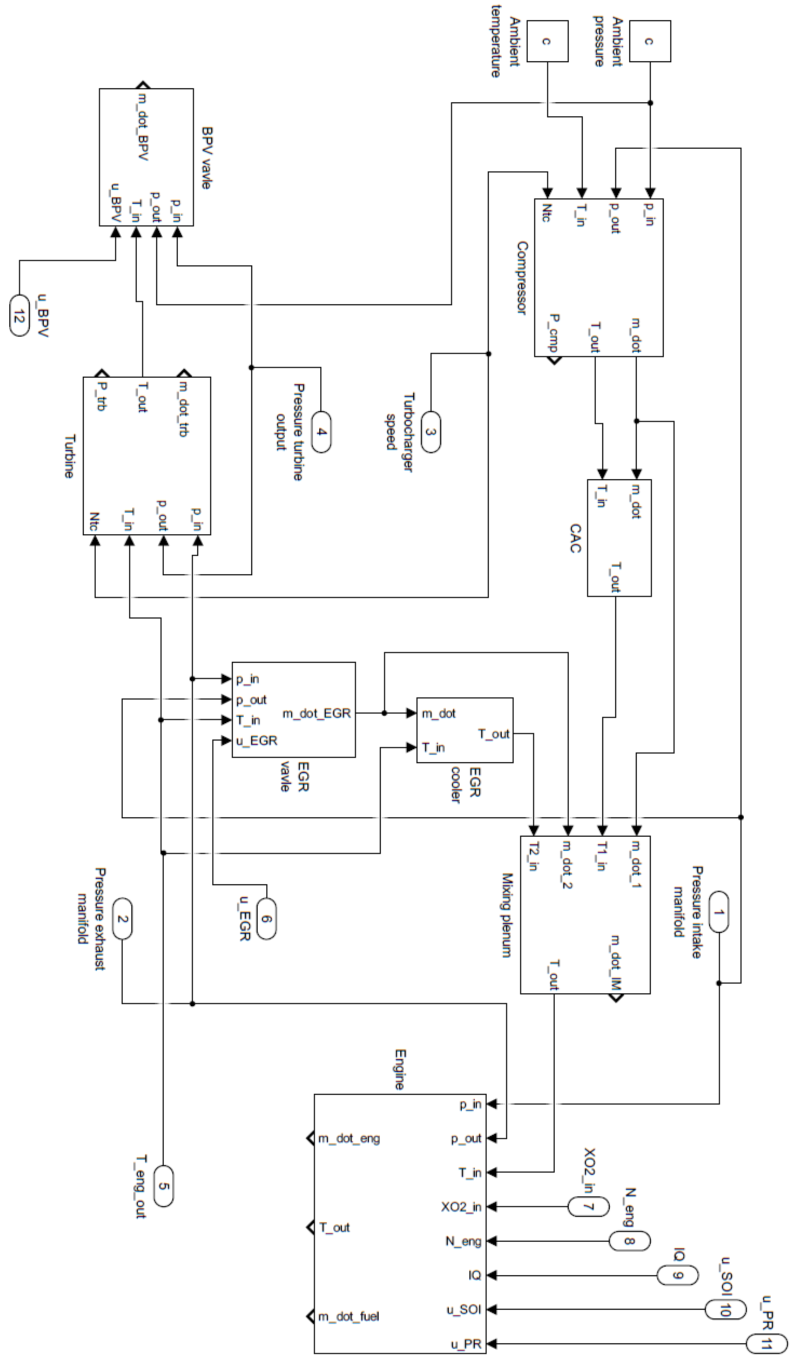

Figure 3.4.1: Input/output ICE model review according to a schema presented in Fig. 3.1.1.

Something deeply hidden had to be behind things.

Albert Einstein

4
Regularized Component Concept

This chapter demonstratively shows how the component regularization can be used to ensure numerical stability of the calibration process and consistency of model signals. First, let us look at the problems that typically arise in steady-state calibration.

Model calibration can be difficult when the model components or model structure implicitly assume that model internal signals satisfy certain conditions, e.g. the positiveness of concentrations. Sometimes it is difficult to find initial model parametrization and initial conditions, which would start model simulation from feasible values. Simulation from infeasible values can make the numerical properties of model calibration problematic. The model Jacobians can be ill-conditioned close to singularities, model can be unstable or even result in finite escape time. The following text will focus on how the model calibration can be approached in such situations.

4.1 Problems of Steady-State Calibration

During a steady-state calibration by finding optimal values of model parameters with an iterative algorithm (mostly a linear or non-linear least squares methods), it can very easily happen that the solver numerically crashes and this can be caused for several reasons:

- *Model singularities* – if the model or some part of it contains a singularity, an iterative algorithm can get into this dangerous part. These algorithms, due to the fast-growing derivatives in this area, will cause the solver to return too high sensitivity values, which usually causes the algorithm to take small steps in iterations and vice-versa. As a result, it can reach an area, where the model no longer represents reality, and the entire calibration process crashes.

- *Poorly selected initial condition* – an important part of each calibration is a good choice of the initial condition. If the initial condition is close to local optima, the calibration process is mostly very successful. However, if the initial condition is way wrong, the process of calibration may not terminate at all.

- *Poorly selected parameter values or internal variables* – in the initial steps of calibration, due to a poorly selected iteration value or a poor input signal (see below), some internal variables becomes non-physical (for example, because of the high measurement noise, the pressure difference of the valve became negative and the air flows through the valve in the opposite direction, for which the model was not designed). As a result, the calibration crashes immediately, or the model produces non-physical outputs.

- *Poor values of component inputs and/or outputs* – the situation is similar to the poor parameter values. Inputs of the model reflect inputs of the real system, as well as their physical properties, which can be limited (temperature > 0 K, pressure > 0 Pa, ...). For example, during the calibration process due to an inaccurately computed vector of parameters, some component of

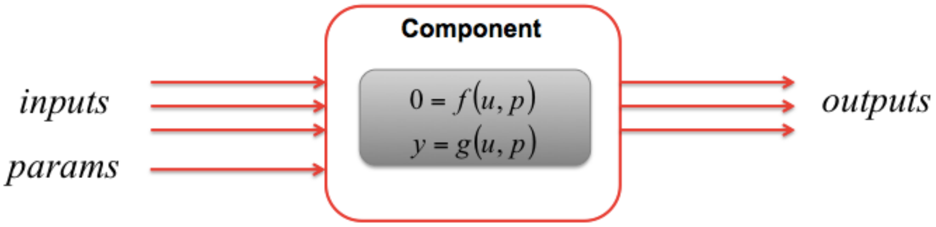

Figure 4.1.1: Steady-state model of component-grey part represents model nonlinear equations. Without the application of internal variables, the structure can not be changed, the simulation and calibration process is not separated in any way.

the model may get a non-physical signal to its input from the previous component and the entire calibration process can fail. This situation may also occur, for example, due to the high measurement noise.

These four situations are the most common cases of the wrong or fatal run of the calibration process. Some of these problems are usually solved in practice by so-called clamping – the hard limitation of the non-compliant parts directly in the model. This solution, however, is quite inappropriate inappropriate when using calibration gradient algorithms, because in the clamped area there is a zero derivative of the model function and hence the zero "extent" within the sensitivity of this function to the parameters sought.

4.2 Regularized Component Concept

Following the above, a concept of regularized component means a component which has its inputs, outputs and internal variables constrained to limited domains as needed, has a well-defined initial calibration condition and is numerically robust for the calibration process.

For this reason, it is necessary to add safety parts to the component that will protect the component from any adverse effects that could disrupt the model calibration. All these model adjustments are taken into account only in case of model calibration, during model simulation model is simulated without any safety parts.

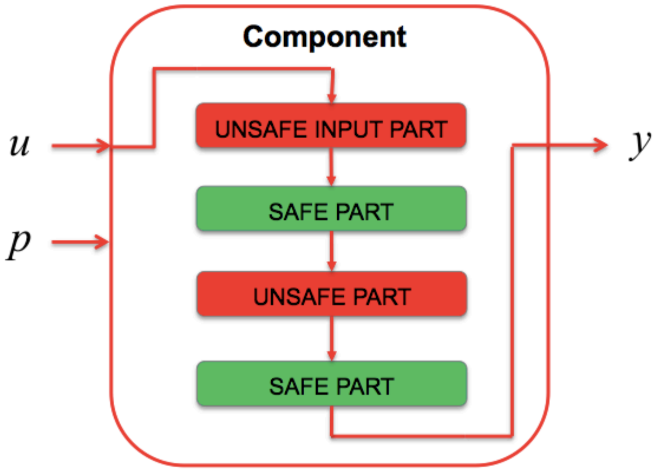

Figure 4.1.2: Steady-state component model. The signal goes through the numerically safe and numerically unsafe parts of the component. Here, for sake of clarity, numerically unstable parts are separated into unsafe input part related to maintain the span of input signals at a physically relevant intervals and into the intern unsafe part which is related to the need of preserve the physical relevance of the component model.

For this reason, the component is divided into two modes – simulation and calibration mode.

4.2.1 Limited Domain Construction

The proposed approach focuses on functions f, g (1.1, 1.2), which were derived from first principles and which can have limited domains

$$(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}) \in \Omega. \tag{4.1}$$

These functions should be differentiable in a closed set Ω and they are supposed to represent the correct physics there.

First example focuses on the oxygen concentration in the combustion products which can be evaluated as a simple fraction, provided both numerator and denominator are positive. For negative values, the formula does not represent reality

$$\begin{aligned} [O_2]_o &= (\dot{m}_i[O_2]_i - \xi\dot{m}_{\text{fuel}})/(\dot{m}_i + \dot{m}_{\text{fuel}}), \\ \Omega &= \{\dot{m}_i[O_2]_i \geq \xi\dot{m}_{\text{fuel}},\ \dot{m}_{\text{fuel}} \geq \varepsilon\}. \end{aligned} \tag{4.2}$$

Another example shows a simple compressor flow function which can fail during calibration phase due to model's discrepancy from reality. Model inputs are input density ρ_{in}, compressor speed N, input pressure p_{in} and output pressure p_{out}. Vector of parameters has five elements, $\mathbf{p}^T = \left[\begin{array}{ccccc} p_1 & p_2 & p_3 & p_4 & p_5 \end{array}\right]$. First four elements of parameter vector are used to calibrate reduced swept volume V_{rs} and the last one is used to calibrate a transportation ratio λ_d.

$$\begin{aligned} \sigma &= p_{\text{out}}/p_{\text{in}}, \\ V_{\text{rs}} &= \mathbf{p}^T \cdot \left[\begin{array}{ccccc} N^3 & N^2 & N & 1 & 0 \end{array}\right]^T, \\ \lambda_d &= 1 - p_5 \cdot (\sigma - 1), \\ \dot{m} &= \rho_{\text{in}} \cdot V_{\text{tr}} \cdot N \cdot \lambda_d, \\ \Omega &= \{\sigma > 1,\ p_5 < 1/(\sigma - 1)\}, \end{aligned} \tag{4.3}$$

where σ represents the pressure ratio, and $\dot{m}$ is a mass flow through the compressor. This model is designed for one direction of flow only, so if $\dot{m} > 0$ then $\lambda_d > 0$ because all another flow formula terms are physically conditioned to be positive. Negative flow can occur due to bad input data ($\sigma = p_{\text{out}}/p_{\text{in}} < 1$) or due to bad parameter constrain ($p_5 \geq 1/(\sigma - 1)$).

Outside Ω functions f, g represent some formal continuation of the physical formulae, but the model calibration can become difficult outside Ω. The possibility of exploring exterior of Ω also increases the risk of finding incorrect local optimum. In general, f, g can contain fractions of functions, square roots of certain nonlinear expressions, etc. It can be usually proven that the fraction denominators and square root arguments in physical formulae must be positive. This defines certain constraints on internal signals.

Proposed approach formulates the steady-state calibration as a constrained optimization problem with feasible iterates [14], [38] to ensure the model variables

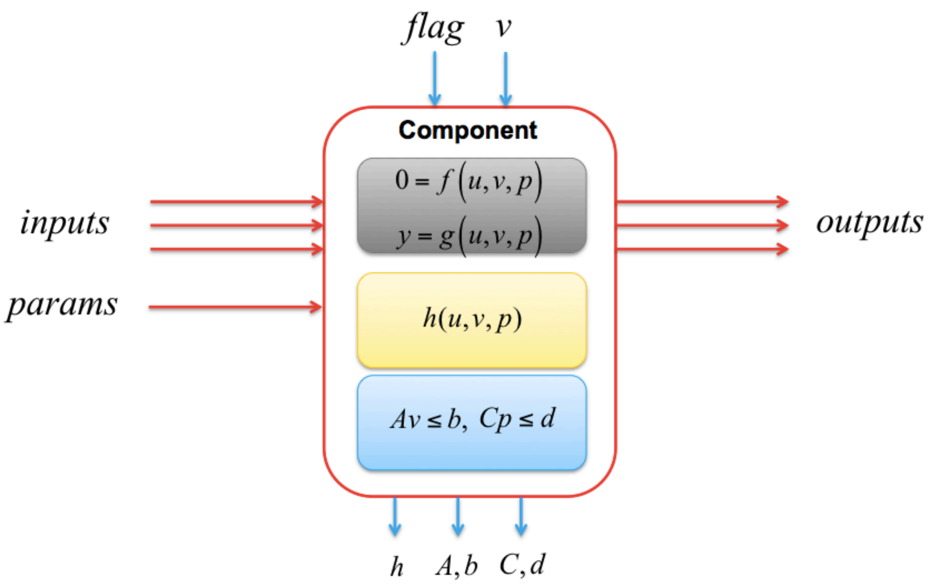

Figure 4.2.1: Steady-state model component with regularization. Grey part represents model nonlinear equations, yellow part represents nonlinear equality constraints and blue part represents linear equality constraints. Variable flag 0,1 switch the component between simulation (classical component approach $= 0$) and calibration (regularized component approach $= 1$) mode.

will stay within the closed set Ω during the whole iteration process, i.e. f, g will not be evaluated for infeasible values during the iterations. The constraints for any internal model signals, functions of states, inputs, and parameters will be allowed. This should lead to fast convergence, because there is no risk of falling outside Ω, when making optimization steps aggressively. The risk of ending in a non-physical local optimum is thereby mitigated.

In Figure 4.2.1 the typical structure of model component regularization can be seen. Every model component has its standard inputs and outputs. In cases where component needs to be regularized it has special inputs to incorporate a new variable v and consequently corresponding special outputs (h) which reflects differences between v and a part, which v replace, see Figure 4.2.2.

The v variables are taken into account only during model calibration process, and replace appropriate model parts in model evaluation. The goal is that these v variables can be linearly constrained for each model component according to

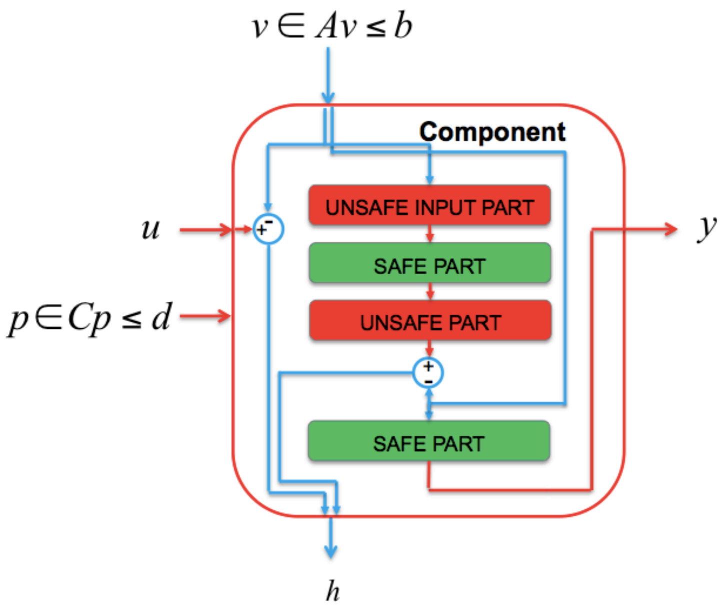

Figure 4.2.2: Steady-state model component with regularization.

its physics or model requirements – this means that the user can explicitly assign the region of component model validity. The differences between component signals and these "fictive" signals will be reflected in the whole cost function (5.4) during the model calibration.

4.3 Model Regularization Example

Typical example of model component regularization can be seen on the model of simple valve component. Model (3.1) is based on first principles and is valid only for case $p_{\text{in}} - p_{\text{out}} > 0$ because of square root function in flow equation. To ensure this constraint, both pressure inputs (p_{in} and p_{out}) can be replaced with new variables (v_1 and v_2) so the model can be regularized. These variables represents pressures which are constrained ($v_1 - v_2 \geq \varepsilon$), so the model will be valid during the whole calibration process. This constraint with another linear constraint to the pressures will be added to a constraint set for subproblems ($Av_k \leq b$).

A similar problem occurs with model parameters. Parameter p_1 represents the valve open cross-section area and parameter p_2 represents the valve closed cross-section area. Logically it must be true that $p_1 \geq p_2$, but during the calibration process there may be situations where this condition may not be met if their box-constraints have a common intersection. Because the global optimizer is the direct owner of the parameter constraints ($Cp \leq d$), there is no problem adding this linear constraint ($p_1 \geq p_2 + \varepsilon$) to this set.

4.3.1 Implementation of a Component Regularization

Let us look at how the regularization of a component quite simply can be implemented in the Matlab environment. The valve flow model (3.1) mentioned in the previous section will be used as an example. All physical quantities are in SI units.

First, the implementation of the component without any modification is introduced[1].

```
function m_dot = valve_model(p_in, p_out, T_in, u_EGR, p)
    % params ->  p(1) open_valve_area, p(2) closed_valve_area
    % inputs -> p_in, p_out, T_in, u_EGR

    R = 288.4 ; %gas constant
    rho = p_in / (R * T_in);
    dp = p_in - p_out;
    A_u = u_EGR * (p(1) - p(2)) + p(2);
    m_dot = sqrt(2*rho*dp) * A_u;
end
```

First of all, all problematic parts have to be modified to separate the simulation and calibration part. After adjusting the problematic parts, there is still a need to add constraints that will be used during the calibration.

[1] For simplicity, the original nonlinear function of the valve opening curve was replaced by a linear one.

The resulting component will look like this:

```
function [m_dot, h] = valve_model(p_in, p_out, T_in, u_EGR, ...
                                  p, flag, v)
    % params ->  p(1) open_valve_area, p(2) closed_valve_area
    % inputs -> p_in, p_out, T_in, u_EGR
    % addit. inputs -> flag, v(1) slack var. in pressure
    %                        v(2) slack var. out pressure

    R = 288.4 ; %gas constant

    if(flag == 1)
        rho = v(1) / (R * T_in);
        dp = v(1) - v(2);
    else
        rho = p_in / (R * T_in);
        dp = p_in - p_out;
    end

    A_u = u_EGR * (p(1) - p(2)) + p(2);
    m_dot = sqrt(2*rho*dp) * A_u;

    if(flag == 1)
        h = [v(1) - p_in; v(2) - p_out];
    else
        h = [ 0; 0];
    end
end
```

Here output *h* reflects the differences between the slack variable *v* and the input pressures, see Figure 4.2.1 and 4.2.2. These errors will be used in calibration on system level as part of the cost function.

An integral part of the component's regularization are also the added constraints to slack-variables and parameters. These can either be part of a component or can be supplied with separate functions as follows.

```
function [Av, bv] = local_inequality_constraints()
    % v_1 -> input pressure [atm]
    % v_2 -> output pressure [atm]
    %   1 <= v_1 <= 7
    %   1 <= v_2 <= 6
    %   v_1 - v_2 > 0.1
    Av = [1 0; 0 1; -1 0; 0 -1; -1 1];
    bv = [  7;   6;   -1;   -1; -0.1];
end

function [Ap, bp] = params_inequality_constraints()
    % p_1 -> open cross section
    % p_2 -> closed cross section
    %    1 <= p_1 <= 100
    % 0.01 <= p_2 <= 10
    % p_1 >= p_2 + 0.1
    Ap = [1 0; 0 1; -1 0; 0 -1; -1 1];
    bp = [100;  10;   -1;-0.01; -0.1];
end
```

Mathematics is the cheapest science. Unlike physics or chemistry, it does not require any expensive equipment. All one needs for mathematics is a pencil and paper.

George Pólya

5
The Algorithm

The optimization method to solve the constrained steady-state calibration is developed in this section.

5.1 Constrained Least Squares for Distributed Calibration

The usual approach to steady-state calibration might involve elimination of the state vectors $\mathbf{x}_k$ at each steady-state operating points solving $f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}) = 0$ with respect to $\mathbf{x}_k$ and locally approximating this this solution with $\mathbf{p}$ as a parameter. The local approximation could be derived from first order expansion around the current iterate $\mathbf{x}_k^{(i)}, \mathbf{p}^{(i)}$

$$\mathbf{x}_k = \mathbf{x}_k^{(i)} - \left(\frac{\partial f(\mathbf{x}_k^{(i)}, \mathbf{u}_k, \mathbf{p}^{(i)})}{\partial \mathbf{x}} \right)^{-1} \frac{\partial f(\mathbf{x}_k^{(i)}, \mathbf{u}_k, \mathbf{p}^{(i)})}{\partial \mathbf{p}} (\mathbf{p} - \mathbf{p}^{(i)}) \quad + \dots \tag{5.1}$$
$$\dots + \quad \mathcal{O}\left((\mathbf{p} - \mathbf{p}^{(i)})^2 \right).$$

Substituting (5.1) into (1.4) turns the optimization problem into optimization with respect to $\mathbf{p}$ only. The proposed approach is based on similar elimination which also approximates $\mathbf{x}_k$ by a linear function of $\mathbf{p}$. As opposed to (5.1), the approximation will be derived from the optimality conditions of a constrained optimization problem. Suppose Ω is described by set of linear and nonlinear inequalities

$$\Omega: \quad \mathbf{A}_p \mathbf{p} \le \mathbf{b}_p, \ \mathbf{A}_x \mathbf{x}_k \le \mathbf{b}_x, \ h(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}) \le 0. \tag{5.2}$$

Equality constraints are not included in the Ω definition, since this kind of problem does not generate such a constraint. The first step is linearization of the nonlinear inequality constraints introducing slack variables $\mathbf{v}_k$, one variable per nonlinear inequality

$$\mathbf{v}_k \le 0, \quad \mathbf{v}_k = h(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}). \tag{5.3}$$

The proposed least squares optimization is formulated with linear inequalities and considering all equality constraints as soft constraints. The diagonal weight matrices $\mathbf{W}_f$, $\mathbf{W}_g$ and $\mathbf{W}_h$ with positive coefficients are used to normalize the problem and prioritize the soft constraints

$$\min_{\mathbf{p}, \mathbf{v}_k, \mathbf{x}_k} \sum_{k=1}^{K} \left\| \begin{matrix} \mathbf{W}_f f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}) \\ \mathbf{W}_g (g(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}) - \mathbf{y}_k) \\ \mathbf{W}_h (h(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}) - \mathbf{v}_k) \end{matrix} \right\|_2^2, \quad \begin{aligned} \mathbf{A}_x \mathbf{x}_k &\le \mathbf{b}_x, \\ \mathbf{A}_p \mathbf{p} &\le \mathbf{b}_p, \\ \mathbf{v}_k &\le 0. \end{aligned} \tag{5.4}$$

The purpose of inequality constraints linearization is having the guarantee of feasible iterates; i.e. no iterate will violate the linear constraints once a single feasi-

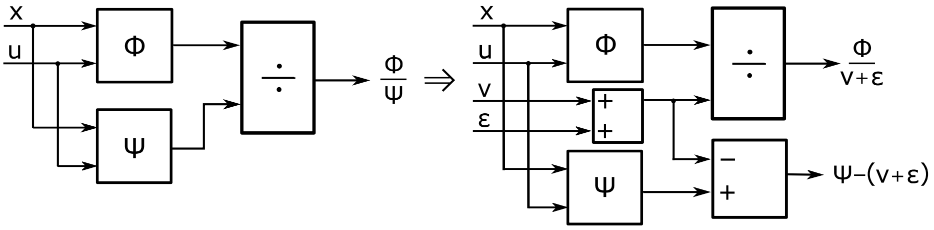

Figure 5.1.1: Model structure modified with slack variables.

ble point is found. To achieve this, the constrained functions $h(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p})$ have to be replaced with $\mathbf{v}_k$ in model functions f, g. In this way the model equations are regularized. As an example: let the model functions contain a fraction of two positive nonlinear functions $\varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p})/\psi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p})$. To regularize this fraction, a non-negative slack variable $v_k = \psi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}) - \varepsilon$ has to be defined; with ε being a threshold. Then the fraction has to be written as $\varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p})/(v_k + \varepsilon)$ in the model equations which will guarantee no division by a number less than ε will occur. This is equivalent to model structure modification as in Figure 5.1.1. For such a modified model, a feasible starting point can be easily found because of linearity of all the inequality constraints. The increased number of the optimized variables is the cost for this augmentation. Introducing an augmented state vector $\mathbf{z'}_k = (\mathbf{x'}_k, \mathbf{v'}_k)$, the problem can be formulated in a compact way. In this formulation

$$\min_{\mathbf{p},\mathbf{z}_k} \sum_{k=1}^{K} \left\| \; \mathbf{F}(\mathbf{z}_k, \mathbf{u}_k, \mathbf{p}) \; \right\|_2^2, \qquad \begin{aligned} \mathbf{A}_z\mathbf{z}_k &\leq \mathbf{b}_z, \\ \mathbf{A}_p\mathbf{p} &\leq \mathbf{b}_p. \end{aligned} \tag{5.5}$$

5.2 Gauss-Newton Method

This regularized nonlinear least squares problem (5.5) with linear inequality constraints will be solved iteratively by a variant of Gauss-Newton method [16] linearizing the function $\mathbf{F}$ at every iteration around the current estimates $\mathbf{z}_k^{(i)}, \mathbf{p}^{(i)}$

$$\min_{\mathbf{p},\mathbf{z}_k} \sum_{k=1}^{K} \left\| \mathbf{M}_k^{(i)}\mathbf{z}_k + \mathbf{N}_k^{(i)}\mathbf{p} - \mathbf{n}_k^{(i)} \right\|_2^2, \qquad \begin{aligned} \mathbf{A}_z\mathbf{z}_k &\leq \mathbf{b}_z, \\ \mathbf{A}_p\mathbf{p} &\leq \mathbf{b}_p, \end{aligned} \tag{5.6}$$

with the sensitivity matrices $\mathbf{M}_k^{(i)}$ and $\mathbf{N}_k^{(i)}$ being Jacobian matrices evaluated at the current estimates

$$\mathbf{N}_k^{(i)} = \left.\frac{\partial \mathbf{F}(\mathbf{z}_k, \mathbf{u}_k, \mathbf{p})}{\partial \mathbf{p}}\right|_{\mathbf{z}_k^{(i)}, \mathbf{u}_k, \mathbf{p}^{(i)}}, \quad \mathbf{M}_k^{(i)} = \left.\frac{\partial \mathbf{F}(\mathbf{z}_k, \mathbf{u}_k, \mathbf{p})}{\partial \mathbf{z}}\right|_{\mathbf{z}_k^{(i)}, \mathbf{u}_k, \mathbf{p}^{(i)}},$$
$$\mathbf{n}_k^{(i)} = -\mathbf{F}(\mathbf{z}_k^{(i)}, \mathbf{u}_k, \mathbf{p}^{(i)}) + \mathbf{N}_k^{(i)} \mathbf{p}^{(i)} + \mathbf{M}_k^{(i)} \mathbf{z}_k^{(i)} \tag{5.7}$$

For sake of clarity the iteration index (i) will be omitted below, assuming that the linearization was made around the last estimates. For complex models, the Jacobians can be approximated by the method of finite differences. Let us analyze the optimal $\mathbf{z}_k$ at an operating point k for a fixed $\mathbf{p}$ using the optimality conditions for this problem

$$\min_{\mathbf{z}_k} \|\mathbf{M}_k \mathbf{z}_k + \mathbf{N}_k \mathbf{p} - \mathbf{n}_k\|_2^2, \quad \mathbf{A}_z \mathbf{z}_k \leq \mathbf{b}_z. \tag{5.8}$$

The Karush-Kuhn-Tucker conditions [5] for (5.8) state that the cost function gradient with respect to $\mathbf{z}_k$ at the optimal point is a combination of the constraints with vector of non-negative Lagrange multipliers μ_k

$$\frac{\partial}{\partial z_k} \|M_k z_k + N_k p - n_k\|_2^2 = M_k' M_k z_k + M_k' N_k p - M_k' n_k = -A_{z,k}' \mu_k, \tag{5.9}$$
$$A_{z,k}^+ z_k = b_{z,k}^+, \quad \mu_k^+ \geq 0,$$
$$A_{z,k}^- z_k \leq b_{z,k}^-, \quad \mu_k^- = 0.$$

The matrix $A_{z,k}^+$ consists of the so-called active rows. The matrix $A_{z,k}^-$ consists of the remaining rows of inactive constraints. Let the set of indices of the active rows is $\mathcal{A}_{z,k}^+$ and the set of indices of inactive rows is $\mathcal{A}_{z,k}^-$. Then sub-vectors μ_k^+ (μ_k^-) are the corresponding active (inactive) elements from the vector of Lagrange multipliers. Using (5.9), one can express the Lagrange multipliers μ_k^+ and the minimizer

z_k in terms of an affine function of p. Defining an auxiliary matrix Q_k

$$Q_k = \begin{bmatrix} M_k' M_k & -A_{z,k}^{+\prime} \\ A_{z,k}^{+} & O \end{bmatrix}^{-1} . \tag{5.10}$$

The inversion of a singular matrix (5.10) may happen when the active inequality constraints are linearly dependent or the M_k matrix columns are not independent. In such case, the matrix Q_k can still be obtained after a simple regularization. Lagrange multipliers and the optimal vector z_k are then the following affine functions of the model parameters p

$$\begin{bmatrix} z_k \\ -\mu_k^{+} \end{bmatrix} = Q_k \left(\begin{bmatrix} -M_k' N_k \\ O \end{bmatrix} p + \begin{bmatrix} M_k' n_k \\ b_{z,k}^{+} \end{bmatrix} \right) . \tag{5.11}$$

Substituting affine functions (5.11) to the inequalities $\mu_k^{+} \geq 0$ for all active constraints and $A_{z,k}^{-} z_k \leq b_{z,k}^{-}$ for all inactive, the set of inequalities (5.12) is defined. Each inequality in $A_z z_k \leq b_z$ produces one inequality in (5.12). The inequalities (5.12) define the ranges for the vector p which define constant set of active constraints on z_k at the operating point k

$$R_k Q_k \begin{bmatrix} -M_k' N_k \\ O \end{bmatrix} p \leq \begin{bmatrix} b_{z,k}^{-} \\ o \end{bmatrix} - R_k Q_k \begin{bmatrix} M_k' n_k \\ b_{z,k}^{+} \end{bmatrix} , \tag{5.12}$$
$$R_k = \begin{bmatrix} A_{z,k}^{-} & O \\ O & I \end{bmatrix} .$$

Here R_k is an auxiliary matrix to simplify the expression. In the following, the inequalities (5.12) will be abbreviated to $A_{p,k} p \leq b_{p,k}$ and the corresponding active rows indices $\mathcal{A}_{p,k}^{+}$. Collectively for all k, these inequalities (5.12) represent the domain of p in which the cost function (5.6) can be locally represented as a sum of norms of affine functions of parameters p only; thus removing the dependence on z_k. In this way, z_k have been eliminated from the optimization provided the parameters p will be varied only within these bounds. Later the same step can be repeated if the set of active constraints on z_k is modified.

Respecting inequalities (5.12), the cost function (5.6) will be minimized with respect to p only. The active constraints resulting from this optimization can be understood as follows. First inequalities $A_{p,k}p \leq b_{p,k}$ in (5.12) originate from the inequalities $A^{-}_{z,k}z_k \leq b^{-}_{z,k}$. The activity of the corresponding constraint in $A_{p,k}p \leq b_{p,k}$ indicates that the cost function descent is blocked by the corresponding constraint which should be put to $\mathcal{A}^{+}_{z,k}$ to improve. The remaining inequalities originate from the inequalities $\mu^{+}_{k} \geq 0$ which ensure that the active constraints $\mathcal{A}^{+}_{z,k}$ will remain active. Therefore, activity of the corresponding constraints in (5.12) means the descent direction of the cost function is pointing from this constraint and the corresponding element of $\mathcal{A}^{+}_{z,k}$ should be inactivated. Both cases can be conveniently described as follows: the activity of a constraint in (5.12) means that the corresponding constraint should be put to or removed from $\mathcal{A}^{+}_{z,k}$, thus its activity being toggled. For all elements of $\mathcal{A}^{+}_{p,k}$, the corresponding elements should be interchanged between $\mathcal{A}^{+}_{z,k}$ and $\mathcal{A}^{-}_{z,k}$. Finally, the solution is a global minimizer of (5.6) in case no constraint among (5.12) is active; $\mathcal{A}^{+}_{p,k}$ is empty for $k = 1 \ldots K$. This active set based optimization algorithm can be summarized as follows:

Inputs: Current estimate of the parameters p, z_k and an array of linearized summands (5.8) given by M_k, N_k, n_k.

(A1) Active sets $\mathcal{A}^{+}_{z,k}$ for all k are determined optimizing (5.8) numerically for all operating points one by one and checking the Lagrange multipliers $\mu_k > 0$.

(A2) Active sets $\mathcal{A}^{+}_{p,k}$ for all k are initialized to contain all constraints (5.12).

(A3) The algorithm finishes when the active sets $\mathcal{A}^{+}_{p,k}$ are empty for all k. This cannot happen at iteration one.

(A4) For all operating points $k \in \mathcal{A}^{+}_{p,k}$, the active sets $\mathcal{A}^{+}_{z,k}$ are transformed into constraints on parameters using the formula (5.12).

(A5) The cost function is expressed as a sum of norms of functions of p only. This is done substituting the affine parametric optimizers z_k given by (5.11) to the corresponding summands for all $k \in \mathcal{A}^{+}_{p,k}$ in (5.6).

(A6) The cost (5.6) is optimized with respect to p solving the constrained linear least squares problem. This optimization considers constraints $A_p p \leq b_p$ as well as (5.12). The positive Lagrange multipliers define the active sets $\mathcal{A}^+_{p,k}$ for each k.

(A8) The active sets $\mathcal{A}^+_{z,k}$ are updated based on $\mathcal{A}^+_{p,k}$ inverting the constraint activity for all elements of $\mathcal{A}^+_{p,k}$. This is based on one to one correspondence between $A_z z_k \leq b_z$ and $A_{p,k} p \leq b_{p,k}$.

Go to A3

Outputs: Optimal solution to the linear least squares problem (5.6): $\hat{p}, \hat{z}_1, \ldots, \hat{z}_k$. This solution satisfies the inequality constraints and approximates the equality constraints in the least squares sense.

To solve the nonlinear least squares problem (5.5), this constrained linear least squares algorithm is used iteratively together with a convenient step control method from convex optimization theory. For example the Levenberg-Marqaurdt approach [16] can be used to penalize the second norm of step in both p and z_k more when the above algorithm has not led to cost function decrease. The model linearization may be triggered only when the estimates p, z_k become feasible; satisfying all inequalities. Until then, the matrices M_k, N_k, n_k can be set to zero and the algorithm searches for a feasible point. Feasible optimization methods must be used at steps (A1) and (A6), such as active set based linear least squares. It is to be noted that this optimization scheme optimizes all augmented states z_k and the parameters p without optimizing all the variables simultaneously. The optimization is done with respect to either z_k, $k = 1, \ldots, K$ for a fixed k, p, or with respect to p without considering z_k, $k = 1, \ldots, K$. The algorithm is thus sufficiently effective even for large K (thousands). The presented algorithm was developed based on the lecture notes on distributed optimization techniques [11].

5.3 Algorithm Applications Examples

In this section a simple examples how the algorithm works will be presented. Firstly, the calibration of an affine system cascade is introduced followed by the calibration of a non-linear system using component regularization. At the end the comparison of the speed of the distributed algorithm with the algorithm which tries to solve the task in a undistributed manner will be presented.

5.3.1 Calibration of the Cascade of the Affine Systems

The calibrated cascade of affine systems can be seen in Figure 5.3.1. The problem is unconstrained. As a steady-state data, pairs $u_{1,k}, \; y_{2,k}$ will be available, where output y_2 was measured and some white noise was added as an unmeasured output error. Without measurement of the output y_1, the calibrated system can be expressed by a single equation as follows

$$y_2 = p_3 p_1 u_1 + p_3 p_2 + p_4. \tag{5.13}$$

It can be seen that the paramterization of the calibrated system is too rich; in other words, the parameters are determined ambiguously because they can not be precisely determined from the data - only the pair $p_3 p_1$ and $p_3 p_2 + p_4$ can be determined, which also results from the fact that the combination of affine functions again leads to a affine function.

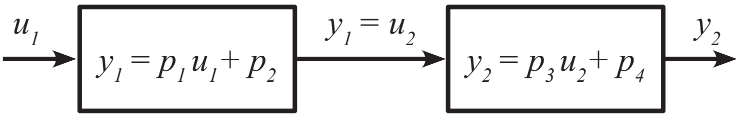

Figure 5.3.1: Calibrated cascade of the affine systems.

Within internal combustion engine calibration, this phenomenon occurs quite often – the parameters are either not uniquely determined or very poorly determined. Even if they can be determined in the deterministic case, the situation is different in the presence of noise. Then some parameter combinations can be extremely sensitive to the noise. Only exceptionally it happens, that all parameters are clearly identifiable. In a more general view, this may mean that although the model well predicts the outputs of the modelled system, it does not mean that the calibrated parameters reflect the true values of the modelled system parameters.

When calibrating this cascade of affine systems, a total of three hundred operating points were used. True parameters values were $p_{\text{true}} = [0.05,\ -0.1,\ 1,\ 0.05]^T$ and initial condition of p was $p^{(0)} = [-1,\ -1,\ -1,\ -1]^T$. Calibration results can be seen in Tab. 5.3.1 and in Fig. 5.3.2.

	p_{true}	$p^{(0)}$	$\hat{p}$
p_3p_1	0.05	1	0.0497
$p_3p_2 + p_4$	0.4	0	0.4012

Table 5.3.1: Result of parameters values within the cascade of affine system calibration.

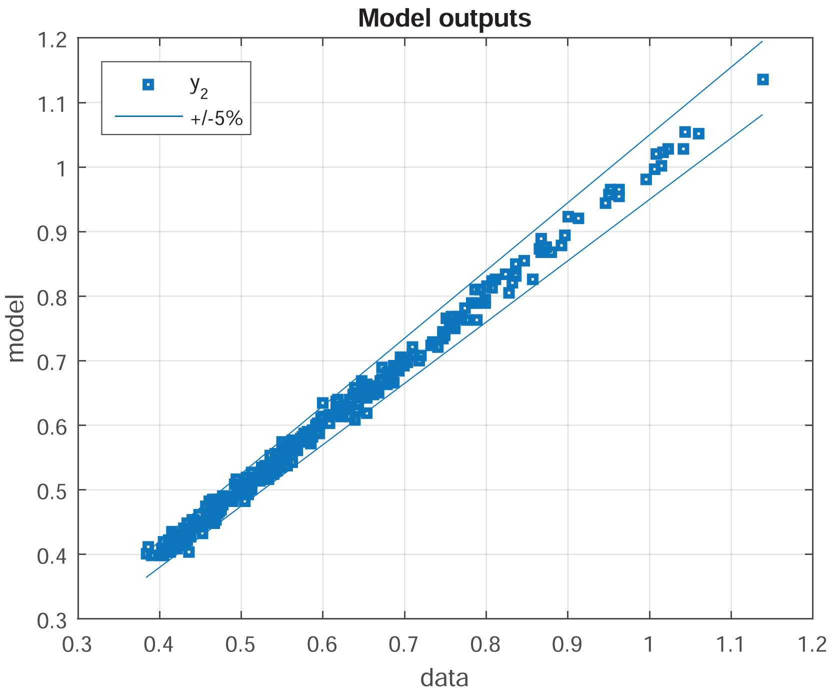

Figure 5.3.2: Calibration results of the cascade of affine systems.

5.3.2 Calibration of the Nonlinear System using Regularization

Now the situation will be changed a bit. From the previous example, instead of the second affine system, a nonlinear system was considered, where the square root function is applied to the input from the previous system. Whole system can be seen in Fig. 5.3.3. Just as in the previous case, only the output y_2 is measured, where some white noise is added as an unmeasured output error. Whole calibrated system can be expressed as follows

$$y_2 = p_3\sqrt{p_1 u_1 + p_2} + p_4. \tag{5.14}$$

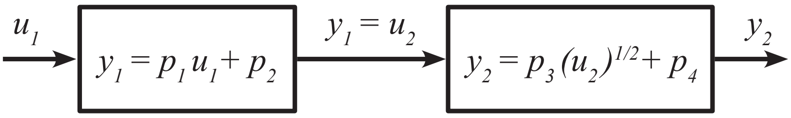

Figure 5.3.3: Calibrated nonlinear system.

In this situation, it is especially necessary to force the input u_2 to be positive. This can be done using regularization of the system. Therefore, it is necessary to supply a slack variable to the second component to ensure that this constraint is satisfied and corresponding slack variable error h, whereby the differences between the u_2 and slack variable are transferred to the minimized criterion

$$u_2 > 0 \rightarrow v > 0,\; h = v - u_2. \tag{5.15}$$

In this example a total of three hundred operating points ($k = 300$) were used. True parameters values were $p_{\text{true}} = [-0.05,\ 5,\ 2,\ 1]^T$ and initial condition of p was $p^{(0)} = [-0.1,\ -0.1,\ 0,\ 0]^T$. Calibration results can be seen in Tab. 5.3.2 and in Fig. 5.3.4 and 5.3.5. If we wanted to calibrate this system using classical methods, the calibration process would usually end up with a solver failure probably due to the complex result of a square root of the negative number.

	p_{true}	$p^{(0)}$	$\hat{p}$
$p_3^2 p_1$	-2	0	-2.1249
$p_3^2 p_2$	20	0	22.1921
p_4	1	0	0.7634

Table 5.3.2: Result of parameters values within the nonlinear system calibration.

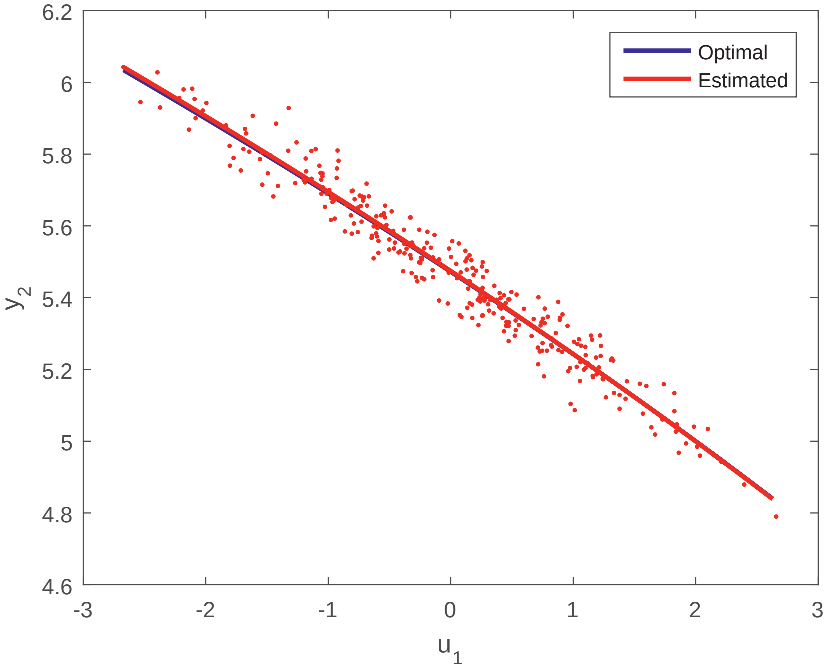

Figure 5.3.4: Affine and nonlinear system calibration result.

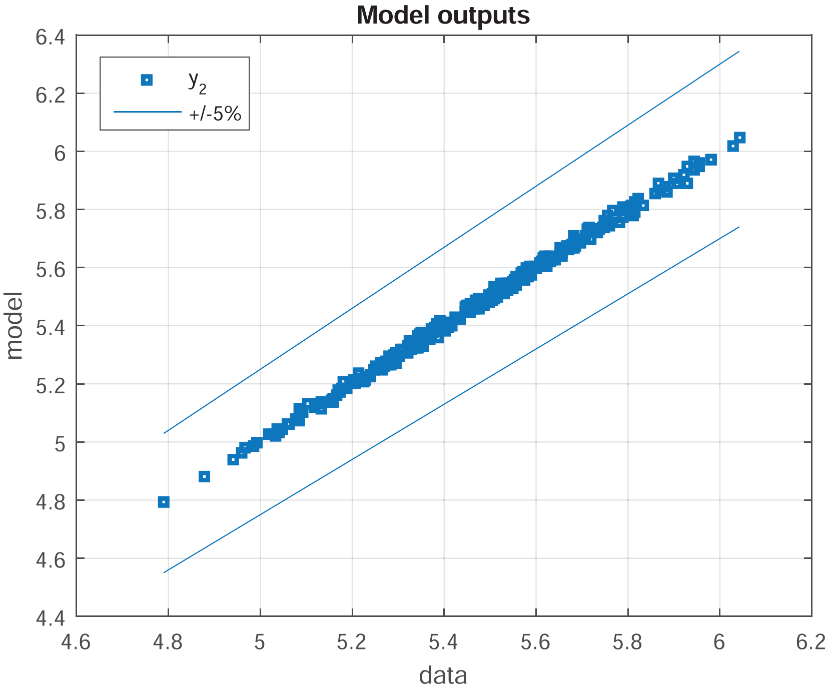

Figure 5.3.5: Affine and nonlinear system calibration result.

5.3.3 Test of the Computational Speed

In the last example the speed of the distributed optimization algorithm compared to undistributed optimization will be demonstrated.

In this example, the QP problem (2.29) was solved for the increasing number of operating points using distributed optimization algorithm presented in Chap. 5 and using interior-point algorithm (undistributed). During both processes, the same number of variables will be used, see table 5.3.3. Used distributed algorithm was implemented in Matlab m-code and interior point algorithm is realized using the quadprog function from Matlab Optimization Toolbox, which does not use sparse matrices during the problem solution.

Number of parameters: 9	Number of linear inequalities for parameters: 4
Number of states: 9	Number of linear inequalities for states: 4
Number of outputs: 11	Parameters initial condition $p^{(0)} = 0$

Table 5.3.3: Problem assignment for computational speed test.

As can be seen in Fig. 5.3.6, from approximately one hundred operating points, the distributed optimization algorithm can solve the task in a faster time with results that do not differ in absolute value from the solution achieved with the interior point algorithm by more than 10^{-5}, see Fig. 5.3.7. Before the number of one hundred of operating points, the distributed optimization has a very similar results as the algorithm being compared within the computational time comparison.

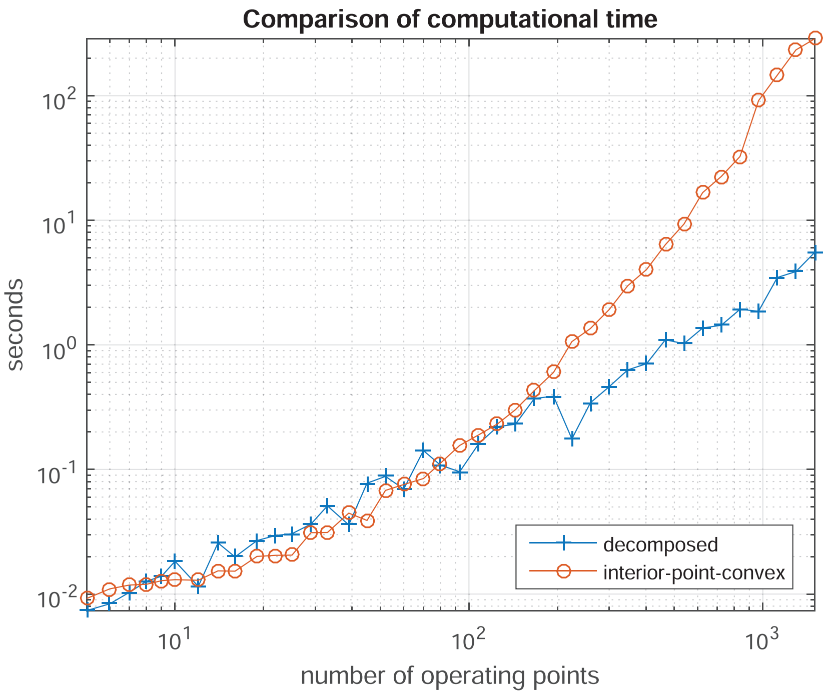

Figure 5.3.6: Result of the test of the computational speed.

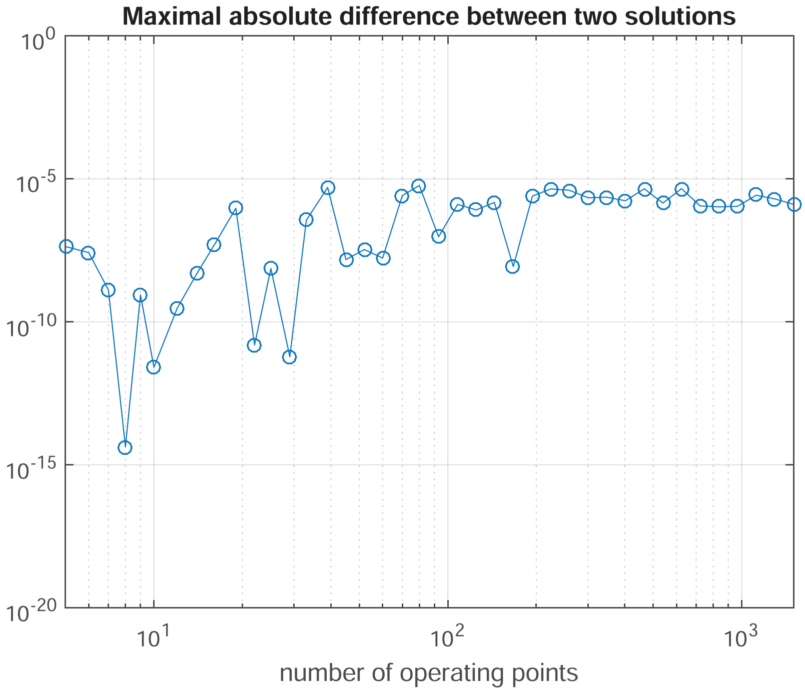

Figure 5.3.7: Result of the test of the computational speed.

It would be possible to describe everything scientifically, but it would make no sense; it would be without meaning, as if you described a Beethoven symphony as a variation of wave pressure.

Albert Einstein

6 Mean Value Model of Internal Combustion Engine

A TURBOCHARGED DIESEL ENGINE is an example of a model which is valid for limited signal ranges. Some limits are defined by singularities. The following paragraphs highlight how these ranges are related to the engine physics. Not all such ranges should not be understood as hard limits. It depends on modeler's decision, whether certain conditions will be modelled, or whether a limit on model validity will be set. A simple MVM is considered which is valid in relatively narrow conditions.

6.1 Turbocharger Nonlinearity

Beside the variable valve timing, engine downsizing with turbocharging is one of the common paths to improved efficiency of modern engines [4]. The turbocharger allows to vary the air density in the intake manifold to a certain extent, which is equivalent to varying the engine displacement. The downsized turbocharged engine can thus behave as a small engine on partial loads where bigger engine would be inefficient. At the same time, the engine is capable of achieving significant peak power through the charged air density increase and thus being able to burn more fuel. The engine can be made bigger or smaller quickly. The speed of the changes is limited only by the turbocharger inertia. This is known as turbo lag. However, modern turbocharged engines exhibit very little turbo lag due to both improved control and decreased moment of inertia of the turbocharger. For these reasons, MVM of turbocharged engines are increasingly important.

From the modelling perspective, even the basic turbocharger represents a highly nonlinear feedback in the dynamical model. A part of the exhaust gas enthalpy is converted to the rotational kinetic energy in the radial turbine. This energy capture is usually controlled partially bypassing the flow through waste-gate or changing the aspect ratio of the turbine blades in variable geometry turbine. The turbine power is used to compress the fresh air in the centrifugal compressor which is installed on the same rotating shaft with the turbine. The air compression modulates the air supply to the combustion process which produces the exhaust gas enthalpy flow. There is clearly a closed loop from the turbine to compressor, and through the combustion process back to turbine. There are several significantly nonlinear multivariate functions in this closed-loop. Calibration of such a model is non-trivial.

The physics based models of both turbine and compressor are usually valid in certain ranges only. With the nonlinearity in the closed-loop, it is hard to guarantee that the initial model parametrization will lead to model signals for which the turbocharger model makes physical sense. That is why the proposed calibration approach is convenient.

From the modelling point of view, the turbocharger is typically calibrated using manufacturer maps [21]. The maps define the corrected mass flows and the efficiencies of both compressor and turbine; a measure of how much power is lost relative to the isentropic process. The corrected flows are defined for normalized inlet pressure and temperature. Flow for general inlet conditions can be then expressed: $\dot{m} = \dot{m}_{\mathrm{ref}}\, p_1/p_{\mathrm{ref}}\sqrt{T_{\mathrm{ref}}/T_1}$.

6.2 Compressor Singularities

At a constant speed, the centrifugal compressor flow decreases with increased compression ratio. At a certain point, the air flow is disrupted, and the compressor stalls. This happens before the point where the mass flow through the compressor gets reversed. The phenomenon is known as compressor surge [31]. The compressor flow becomes unstable, exhibiting oscillations due to locally negative sensitivity of the flow with respect to pressure ratio. Exact modelling of such unstable flow is usually beyond the scope of a simple MVM.

This does not mean that the model would not be useful because the compressor does not normally operate in surge except for short, and usually undesirable, periods of time after an abrupt closing of the throttling plate (tip-off). Some compressors are even equipped with decompression valve which prevents the surge conditions opening a bypass flow line which quickly decreases the compression ratio when the throttle is closed. The surge represents an inequality constraint for the control system which can be controlled without using a model of the surge flow.

On the opposite end, the compressor flow increases with decreased compression ratio up to a point where the flow reaches sonic conditions, when the flow is choked. This changes the flow nature again. Mass flow is saturated and cannot be increased even if the compression ratio would be further decreased.

The normal compressor model is valid only between the surge and choke conditions. Numerically, the surge and choke make the model calibration problematic because the gradients either vanish (choke) or become unbounded (surge). An accidental crossing of the surge limit causes the compressor model to reverse the flow which usually cannot be handled correctly by the rest of the model because the locations of causes and effects are interchanged. Both surge and choke can be approximately located on the compressor map, analyzing the sensitivity of the flow with respect to the pressure ratio as in Figure 7.1.1. The surge condition is close to the point where the sensitivity $\partial \dot{m}_c / \partial (p_2/p_1)$ is unbounded, the choke condition exhibits zero sensitivity. When expressing the flow as a function of pressures, the equation will contain a singularity, division by zero happening just beyond the surge limit. This singularity is already in the area where the model does not reproduce physics.

The centrifugal compressor mechanical power is given as follows [18]:

$$P_c = \frac{1}{\eta_c} c_p \dot{m}_c T_1 \left((p_2/p_1)^{(\gamma-1)/\gamma} - 1 \right) . \tag{6.1}$$

Isentropic efficiency η_c ranges approximately between 0.3 and 0.8 during normal operation. Outside the normal operation ranges, the efficiency falls to zero very rapidly towards both surge and choke conditions. Using equation (6.1) with the efficiency map fitted by e.g. a rational polynomial function leads to division by zero easily. Formally, zero compressor efficiency leads to infinite deceleration of the turbocharger and infinite compressor outlet temperature. This is another manifestation of the fact that the compressor operates in ranges surrounded by singularities in the equations. These singularities define a feedback which does not allow the physical system to escape from Ω. However, the numerical model simulation and calibration are not protected.

6.3 Turbine

The ranges of turbine model validity are similar to the compressor. The flow exhibits unbounded sensitivity with respect to the pressure ratio for zero flows. For increased pressure ratio, the flow chokes and becomes constant. The turbine model is regular only for a certain reasonable range of the pressure ratios.

6.4 Flow Restrictions

The flow restrictions and valves cause the pressure drops of the gas flows in the engine. The following equation based on flow function Ψ is usually used for flows which are not choked [18]:

$$\dot{m} = c_d A \rho_i \Psi \left(p_\mathrm{i}/p_\mathrm{o} \right) , \tag{6.2}$$

$$\Psi \left(p_\mathrm{i}/p_\mathrm{o} \right) = \left(\frac{p_\mathrm{i}}{p_\mathrm{o}} \right)^{\frac{1}{\gamma}} \sqrt{ \frac{2\gamma}{\gamma - 1} \left[1 - \left(\frac{p_\mathrm{i}}{p_\mathrm{o}} \right)^{\frac{\gamma-1}{\gamma}} \right] } . \tag{6.3}$$

This function has infinite derivative for the pressure ratio $p_\mathrm{i}/p_\mathrm{o} = 1$, and should be used only for $p_\mathrm{i}/p_\mathrm{o} > 1$. In our approach, a minimum pressure ratio can be set as a linear constraint: $p_\mathrm{i} \geq p_\mathrm{o} + \varepsilon$.

6.5 Recirculation Loops

Modern engines are equipped with exhaust gas recirculation (EGR). The normal EGR flow direction through the recirculation valve is from exhaust to intake. The energy and mass balances are based on the fact that the recirculated gas temperature starts from the exhaust temperature. When the flow direction changes, sign reversal happens in the equations. However, it should also be assumed that the recirculation temperature is switched from T_3 (exhaust) to T_2 (intake). Thus, there may be a discontinuity of gradients in the equations. Formal (incorrect) use of the equations leads to models which do not satisfy energy conservation. It is probably easier to put a linear constraint on pressures which makes the flow reversal impossible rather than trying to model such phenomena. The flow reversal rarely happens in real engines.

Everything must be made as simple as possible. But not simpler.

Albert Einstein

7 Engine Model Calibration

To demonstrate the calibration method, a highly simplified MVM of a 2.2 l straight four cylinder turbocharged diesel engine was used. The engine schematics is in Figure 7.0.1. The fresh air is compressed by the compressor. The compressed air is cooled by the charged air cooler to increase its density. The cooled air is then mixed with a fraction of cooled exhaust gases into a reservoir of 8 l in order to smooth the boost pressure. The cylinders are filled from this reservoir and the diesel fuel is injected directly into the cylinders. The exhaust gas recirculation is controlled by the Exhaust Gas Recirculation (EGR) butterfly valve. The remaining exhaust gas is expanded in the variable geometry turbine to capture part of its energy. The turbine drives the compressor by the connecting shaft. The state variables include the intake and exhaust pressures and temperatures and the turbocharger speed. In this simplified scheme, a number of flow restrictions and heat transfers have been neglected for simplicity.

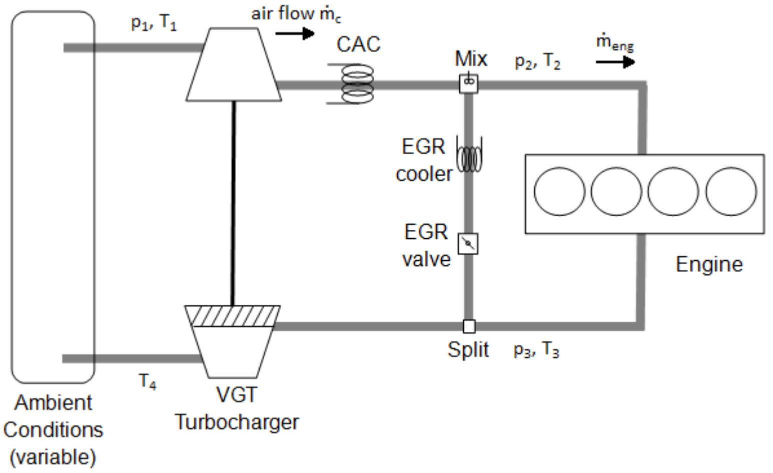

Figure 7.0.1: Turbocharged diesel engine model schematics.

Each model component was modeled by equations briefly explained in the subsequent paragraphs.

7.1 Compressor

The compressor map has been fitted by a model almost identical to the model suggested by [24], see Figure 7.1.1. This model is based on a rational functional representation of the dimensionless flow rate Φ_c as a function of dimensionless head parameter Ψ_c. The coefficients of the rational function are considered to be dependent on the flow Mach number M. Compared to [24] quadratic terms of Mach number were also considered:

$$\Phi_c = \frac{k_3 \Psi_c - k_1}{k_2 + \Psi_c}, \qquad k_i = k_{i,0} + k_{i,1}M + k_{i,2}M^2. \tag{7.1}$$

The dimensionless parameters are defined as follows:

$$\Psi_c = c_p T_1 \left((p_2/p_1)^{(\gamma-1)/\gamma} - 1 \right) / U_c^2, \qquad \Phi_c = 4\dot{m}_c / (\rho_1 \pi d_c^2 U_c). \tag{7.2}$$

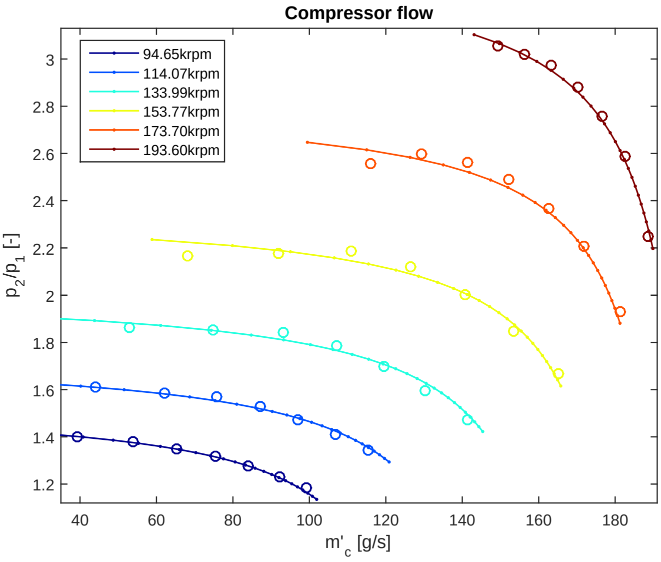

Figure 7.1.1: Compressor flow map and its fit by rational polynomial function.

The isentropic compressor efficiency has been fitted by a rational polynomial function of the dimensionless head parameter and the pressure ratio as in Figure 7.1.2. The figures demonstrate a significant nonlinearity.

7.2 Heat Exchangers

Both charged air and recirculation coolers were modeled by the algebraic equation [18]:

$$T_o = T_i - \eta_{\mathrm{he}}(\dot{m})(T_i - T_{\mathrm{cm}}), \tag{7.3}$$

where T_{cm} is the cooling medium temperature and $\eta_{\mathrm{he}}(\dot{m})$ is heat exchanger effectiveness between zero and one. Supposedly the effectiveness equals one for zero gas flow and tends to zero for flow approaching infinity. Therefore, the effectiveness was represented by $2^{-\dot{m}/a}$, with a parameter a equal to gas mass flow value for which the effectiveness drops to 50 %.

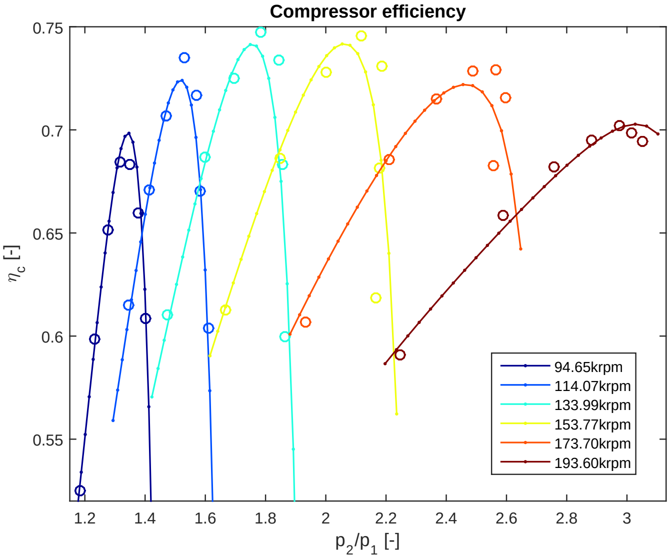

Figure 7.1.2: Compressor efficiency map and its fit by rational polynomial function.

7.3 Combustion Model

The combustion process was modelled to define the charge mass flow and temperature in the exhaust manifold. The engine charge mass flow was represented by the well known formula:

$$\dot{m}_{\text{eng}} = V_d \eta_{\text{vol}} \rho_2 N_e / 120. \tag{7.4}$$

The volumetric efficiency η_{vol} was represented by an empirical function of N_e and the pressure ratio p_3/p_2, see [13]. To keep the model very simple, the combustion process was not modeled in detail. In outline: the thermal efficiency was calculated from the ratio of torque and injection quantity:

$$\eta_{\text{eng}} = \frac{4\pi}{N_{\text{cyl}} H_{\text{LV}}} \frac{T_q}{m_{\text{inj}}}. \tag{7.5}$$

This thermal efficiency estimate was fitted using a quadratic function of air to fuel ratio, injection quantity and engine speed. This function represents the fraction of fuel energy converted to mechanical power. The remaining energy is split between the heat dissipation and the exhaust gas enthalpy. To calculate the exhaust manifold temperature, the heat loss was again approximated by an empirical formula. Then the remaining energy was equated to the exhaust gas enthalpy. The combustion model was not configured to model sub-stoichiometric combustion which defines another constraint on model validity. This limit is associated with a discontinuity in sensitivity. Adding more fuel above the stoichiometric value does not increase the exhaust temperature but decreases it. This simplified combustion model did not use full information about the injection pattern, rail pressure etc. Only the total injected fuel quantity was considered.

7.4 Turbine

The turbine reference flow was fitted to an empirical modification of the orifice equation:

$$\dot{m}_t = C(u_{\mathrm{VGT}}, N_{\mathrm{tc}})\,(p_3/p_4)^{\kappa(u_{\mathrm{VGT}})}\,\sqrt{(p_3/p_4)^{-2/\gamma} - (p_3/p_4)^{-(\gamma+1)/\gamma}}. \qquad (7.6)$$

The discharge coefficient C and the exponent parameter κ were considered to be polynomial functions of the turbine geometry parameter u_{VGT} and speed N_{tc}. The turbine efficiency was fitted by a rational polynomial function of the pressure ratio, turbine speed, and u_{VGT}. Assume $\tilde{u}_{\mathrm{VGT}}[\%]$ is an input to an actuator which rotates the turbine vanes to a position. In the manufacturer turbine map, the VGT position was represented by a dimensionless number u_{VGT} between zero and one. In order to use the turbine map in the MVM, the correspondence between u_{VGT} (map) and $\tilde{u}_{\mathrm{VGT}}$ (ECU) was modelled. A simple affine function equation was used: $\tilde{u}_{\mathrm{VGT}} = k_1 u_{\mathrm{VGT}} + k_0$. The optimization resulted in values almost equal to $1 - 0.01 u_{\mathrm{VGT}}$ which indicates the control signal orientation was opposite but the extreme values of both signals matched the extreme points in the map.

7.5 Model Constraints

This section explains how this MVM validity was represented by a set of linear constraints on model states, slack variables and parameters. The equations defined by the model structure are valid for defined flow directions only. Flow reversal modelling would require reconfiguration of the model equations which may be impractical. To this end, three linear inequalities $p_{\text{amb}} < p_1 < p_2 < p_3$ were introduced. These constraints made the negative flows through the EGR and the turbine impossible. The compressor flow direction constraint is more complicated. It was necessary to introduce a slack variable $\dot{m}_c \geq l(\dot{m}_{\text{fuel}})$, which was used as an input to the rest of the model (the charged air cooler). The limit $l(\dot{m}_{\text{fuel}})$ was set to a value slightly above the stoichiometric air to fuel ratio; considering the limits of validity of the exhaust temperature model used. Thus the model never sees a negative flow value from the compressor equation. If this happens, the optimization tries to correct the rest of the model to change the compressor flow direction decreasing p_2. Positive slack variables were also introduced for all three denominators of rational polynomial functions existing in the model: compressor flow parameter (7.1.1) and both compressor and turbine efficiency.

Additionally all state variables, including the turbocharger speed and the temperatures, were constrained by upper and lower limits. The model parameters were also constrained by box or linear constraints whenever possible. As examples: the discharge coefficient of the EGR valve must be positive, the effectiveness of the coolers must decrease with the mass flows which implies $a \geq 0$ for the heat exchanger etc. The exact configuration of constraints and slack variables was based not only on the knowledge of chief singularities and model validity limits but also on experimentation with the optimization approach.

7.6 Model Calibration

A collection of 768 steady-state points measured on the test cell dynamometer as well as the turbine and compressor maps from the turbocharger manufacturer have been used to calibrate the engine model. Each steady-state point has been measured setting engine speed and injection quantity to certain values and then ex-

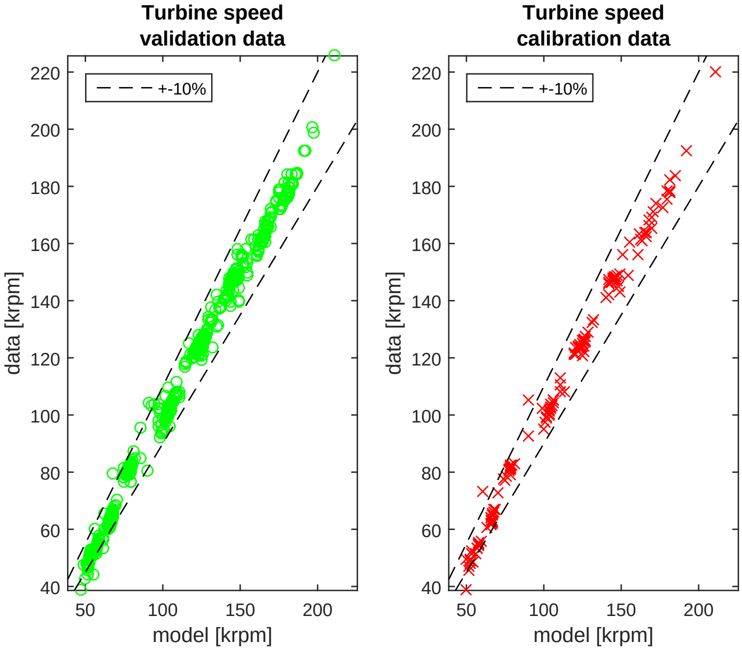

Figure 7.6.1: Turbine speed steady-state identification results.

ploring a grid of EGR valve and VGT actuator positions. All other parameters, e.g. the injection pattern, were set to their normal operating point dependent values. At each point the engine was left 120 seconds running for the temperatures to settle. Then, measurement values were collected from following sensors: air flow, temperature after compressor, temperature after charge air cooler, pressure and temperature in the intake manifold, engine torque, pressure and temperature before turbine, exhaust gas recirculation rate based on CO_2 sensors, lambda sensor located after the turbine, turbine speed. It was ensured that all data were satisfying model validity constraints; thus excluding data with low air to fuel ratio or data with slight negative EGR pressure difference etc.

The model components were calibrated independently to define reasonable initial parameters. In this step, the component-wise calibration approach was used, using a standard nonlinear least squares algorithm. The exceptions were model components expressed as rational polynomials: the compressor flow and both

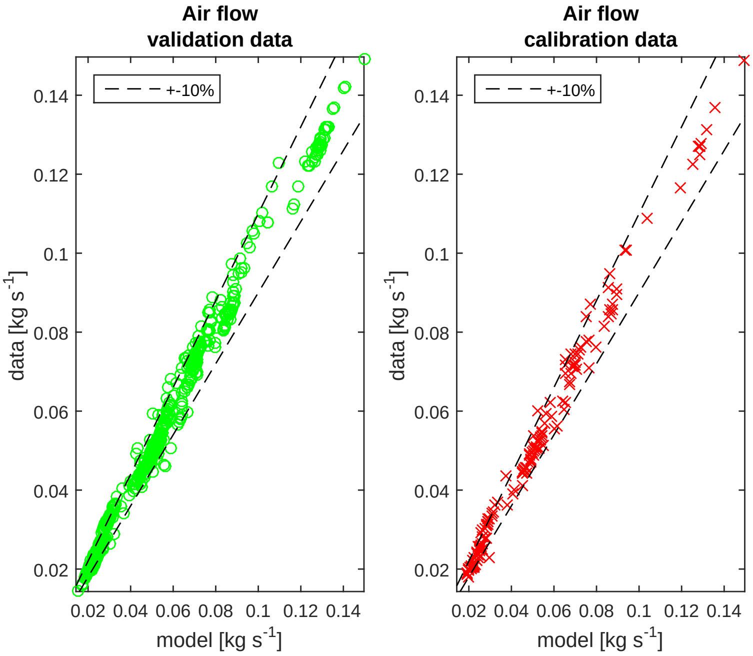

Figure 7.6.2: Fresh air flow steady-state identification results.

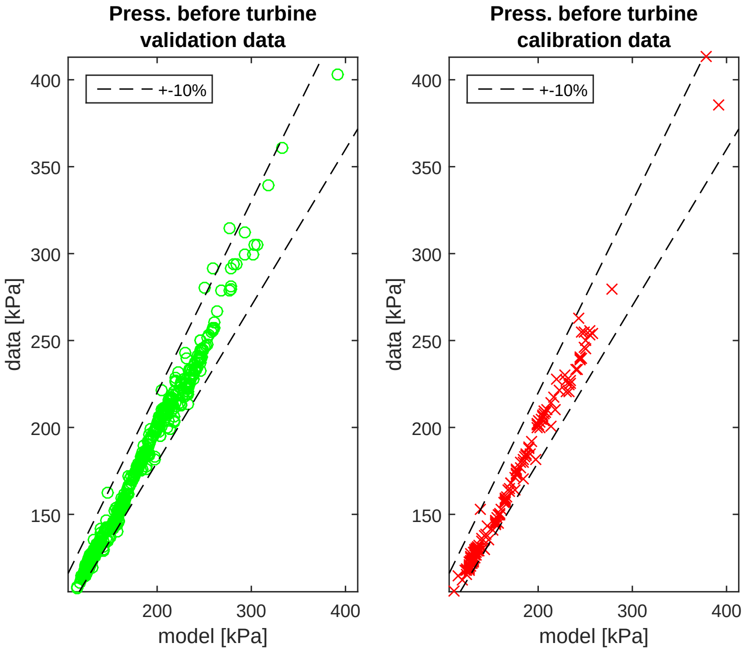

Figure 7.6.3: Pressure before turbine steady-state identification results.

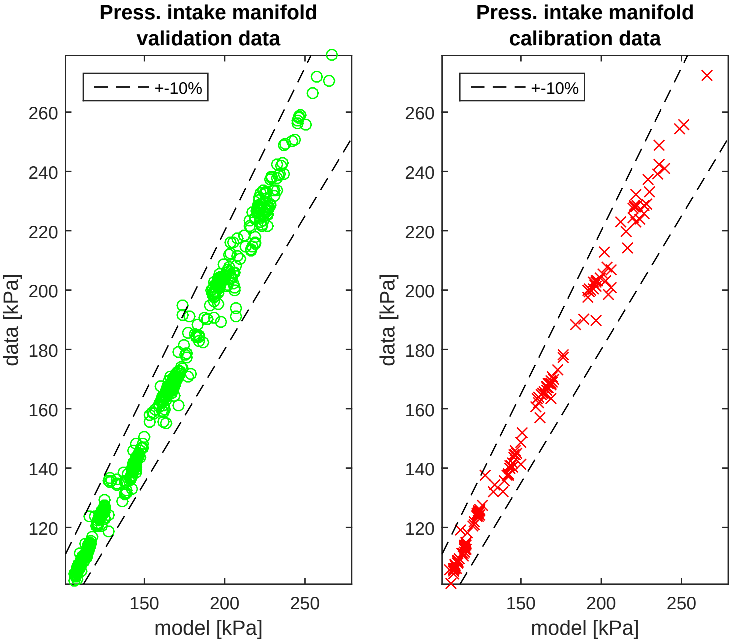

Figure 7.6.4: Intake manifold pressure steady-state identification results.

compressor and turbine efficiency. Here, the proposed algorithm was used introducing constrained slack variables as denominators of the functions. This prevents singularity crossing provided the initial parameter values are set reasonably.

Next, the proposed system level calibration algorithm was used to improve the fit with respect to all sensors. At this point, the focus shifted from individual components to the global model, which includes the causal links between the components. The optimization varied all model parameters. The parameters were box constrained to vary not more than $\pm 20\ \%$ from their initial values. The parameters related to the turbomaps were allowed to vary in an even narrower range of $\pm 5\ \%$. The optimization used only 20 % of the data taking every fifth point. The remaining 80 % data were used for verification only. The model calibration was based on implementing the function **F** from (5.5) as a Matlab function. This function was called iteratively by an algorithm **A** prototype also implemented in Matlab. The partial derivatives of F were evaluated using symmetric finite differences.

7.7 Calibration Results

The model accuracy before and after the system level optimization step is summarized in Table 7.7.1 and visualized in Figures 7.6.1, 7.6.2, 7.6.3 and 7.6.4. The R^2 is the coefficient of determination used as a measure of fit quality. It is defined as squared correlation coefficient between the model prediction and the measurement, including both calibration and verification sets. The optimization based on the proposed algorithm and the Levenberg-Marquardt step control took a few minutes on standard PC and no numerical problems occurred even when testing with randomly varied initial parameters. The calibration could be successfully completed even if it started from initial parameters with which the model could not be simulated for numerical reasons; e.g. defining the initial EGR valve discharge coefficient too large. System level approach improved the fit quality significantly, as expected. The accuracy of the model built from components which were calibrated separately is worse. Figures 7.6.1, 7.6.2, 7.6.3 and 7.6.4 show steady-state model predictions obtained by simulating the calibrated model for four important model variables. These results were obtained solving the differential equations of the model without considering the slack variables. Thus, these results are not affected by the effects of soft constraints formulation. There is no significant difference between the calibration and validation data sets; thus the model was not over-parametrized.

7.8 After the Steady-State Calibration

Though the proposed approach is focused on the steady-state calibration only, the question whether the model calibrated in this way can be useful for transient simulation and controller design is a valid one. In our experience, the MVM with a reasonable steady-state response can be easily adjusted optimizing a subset of parameters to match also the transient data quite well. If only parameters which do not affect the steady state solution are chosen then both problems are decoupled. The parameters to be optimized in this step can be: actuator and sensor delays, heat capacities, turbocharger inertia, intake manifold volume. Transient step re-

signal	R^2 before	R^2 after
press. before turbine	79.6 %	98.4 %
press. intake manifold	81.0 %	98.8 %
turbine speed	90.4 %	99.2 %
lambda	2.6 %	96.9 %
EGR rate	85.6 %	91.2 %
temp. before turbine	68.9 %	97.2 %
torque	98.7 %	99.4 %
temp. intake manifold	0.0 %	81.9 %
temp. after intercooler	83.6 %	93.8 %
temp. after comp.	88.3 %	98.2 %
air flow	81.7 %	99.1 %

Table 7.7.1: Calibration results before and after running global level calibration.

sponse of a model calibrated in this way is shown in Figure 7.8.1. It can be seen that shown system has a non-minimum phase behaviour. ERG valve is situated between intake and exhaust manifold and for proper engine operation, there is a typical pressure difference $p_{EM} > p_{IM}$. Because of this, when opening the EGR valve, an increase of intake manifold pressure will firstly occur. In consequence the pressure in the exhaust manifold begins to decrease, so the turbine power is decreasing too. Thereby the mass flow through the compressor is decreasing which results in the intake manifold pressure decreasing.

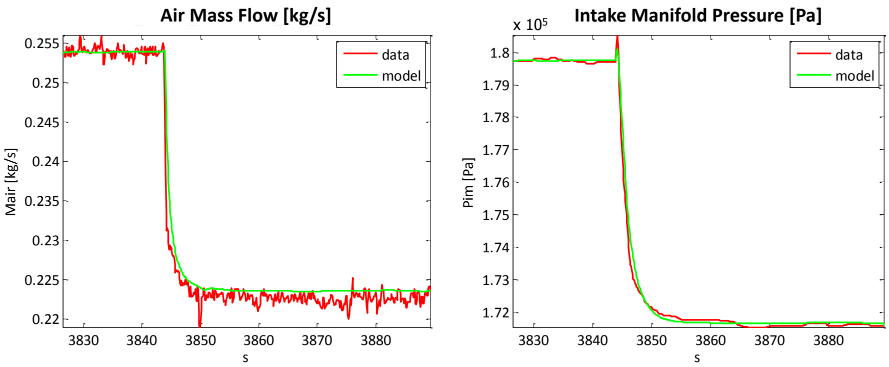

Figure 7.8.1: Transient response of the MVM to EGR valve position step from 20 % to 40 %.

Success is not final, failure is not fatal: it is the courage to continue that counts.

Winston S. Churchill

8 Conclusion

8.1 Summary

This work deals with identification of nonlinear systems in steady-states. Because the structure of the model is explicitly given, the term calibration of the system is used rather than identification.

Prior to developing the distributed calibration method, the calibration of models was based primarily on numerical optimization. However, this optimization has often suffered from numerical problems that were difficult to solve. Optimization should be very conservatively tuned so that only small steps are done during each iteration. In the larger steps of the optimization algorithm, singularity could be maybe "skipped". Also, the initial values of the model parameters must be carefully set. Even then optimization sometimes ended in a solution that had no physical meaning. Sometimes it also happened that the model calibration process resembled an attempt to change something randomly without a proper systematic method. If there was an error in the calibration process, it was very difficult to find the cause and fix it. For complex systems that appear primarily in engineering practice, it is also no exception that these systems consist of multiple subsystems – components. A typical example can be the introduced model of a diesel combustion engine, which is composed of a number of components. These components are then combined into the overall system, and it can be seen that as the correct representation of a particular component model is important, interactions of these components within the overall system are also very important as well.

The solution of these problems is presented systematically in this thesis. Firstly, the methods commonly used to solve steady-state calibration problem (see Chapter 2) are presented. In the end of this Chapter, the principle of the distributed optimization to solve complicated optimization problems is presented. This concept is used further when constructing the calibration algorithm. The Mean-Value Model of a diesel internal combustion engine including all the model components based on the given physical background is introduced in Chapter 3. In Chapter 4, calibration methods and Mean-Value Models are further analyzed in terms of numerical stability of the calibration process. Frequent drawbacks of models are shown, such as insufficient delimitation of the model's validity domain, poor treatment of signals passing through components or poorly selected vector of parameters terms or throughout the calibration process iterations. These shortcomings are then reflected in the wrong calibration process, both at the component level calibration and at the level of the global model calibration. All negative properties are gradually solved in the context of introducing additional component states and introducing additional constraints to these states. This process is called regularization in this work. In Chapter 5, a distributed calibration algorithm is introduced which uses the component regularization concept. The initial concern with the proposed approach was related to the soft constraints in the formulation (5.4). The model has more degrees of freedom during the calibration than simulation because the equalities are not satisfied at the terminal iteration, though the errors should be small setting the weights W_f, W_g, W_h in (5.4) properly. However, it seems that these additional degrees of freedom plays role similar to optimizing errors of independent variables in the Total Least Squares method, see [28]. The hypothesis is that the use of slack variables actually improves the statistical robustness. However, the latest research showed that the algorithm performs better if the equality constraints which become feasible are turned into real equality constraints during the process. It should be noted that the proposed calibration approach cannot improve the model accuracy by itself. The model accuracy is mostly given by model structure which must properly capture the chief physics. The results obtained using different calibration methods with the same model structure and same data should be almost identical provided the global optimum was found. The difference is that the calibration process can be faster and easier. However, this proposed approach is not a universal global optimization method. The estimated parameters can only represent the local optimum. The chapter 6 discusses the problems that occur when calibrating the model of the diesel internal combustion engine. Based on regularization techniques, this model is regularized in Chapter 7 to improve the numerical stability of the calibration process. Within this process, components are firstly calibrated independently using the nonlinear least squares algorithm and then the calibration on a global level is done. The greatest benefit of the distributed calibration using regularization techniques is the numerical stability of the entire calibration process. It is also shown that greater homogeneity of calibration results is achieved within the global calibration process, see Tab. 8.1.1. Last but not least, the calibration process can be faster and easier using the techniques.

8.2 Contributions of the Author

8.2.1 Regularized Component Concept

One of the main contributions of this thesis is the introduction of the concept of regularized component in Chapter 4 and 5, which were presented in [7] and [6]. During the steady-state calibration process of complex nonlinear systems composed of components, a number of numerical problems can occur. The most significant are listed below

- The component in cascade may not receive a relevant physical signal from the component before.

- The component may contain functions with limited domains, e.g. rational

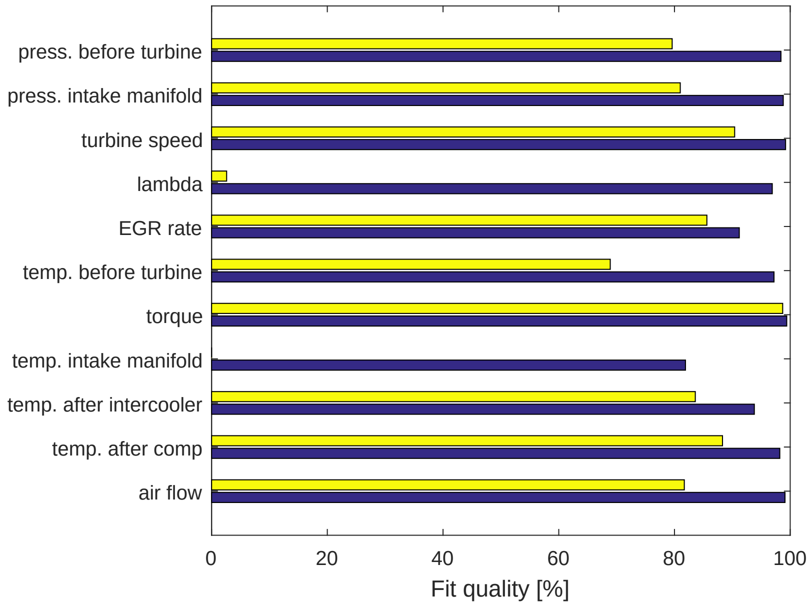

Figure 8.1.1: Graphic interpretation of calibration results before and after running global level calibration from Table 7.7.1. Yellow bars are related to fit quality before global level calibration and blue bars after it.

functions or square roots and so on, which can lead to the calibration process failure.

- Initial condition of the vector of optimized parameters may be poorly selected.
- During the calibration process the vector of parameters is updated in a way which may cause that the model enter a domain where it is not physically correct.

All these problems can lead to a poor results of the calibration process or to a solver numerical failure.

These problems are solved in this work by introducing additional constraints that apply during the calibration process and which are held during the iterations. This includes the introduction of slack-variables that can be viewed as additional degrees of freedom in the model structure. They make the critical causal links between model components less stringent, converting them to soft constraints. The component level calibration approach neglects the causal links between components completely. At the same time, the system level calibration fully relies on these causal links. It follows that the proposed approach can be understood as a generalization of both component and system level calibration approaches depending on the weights put on the soft constraints. It can be described as a reconciliation of model components to meet a set of constraints and minimize the system model predictions at the same time. To my knowledge this idea is novel.

8.2.2 Calibration of Mean-Value Diesel Engine Model

It is commonly assumed that it is sufficient to have precisely calibrated components. This idea is was refuted by the results of the global calibration in Fig. 8.1.1. The calibration of the model on the global level achieved a great improvement in the model prediction accuracy. Furthermore, after the global calibration, the results are much more homogeneous than after the component only calibration. Within the eleven monitored variables, the average coefficient of determination was improved from 69.12 % related to the component fit to 95.82 % related to the fit after global level calibration, see Tab. 7.7.1. This increases the integrity of the calibrated model.

During a calibration process a collection of 768 steady-state points measured on the test cell dynamometer as well as the turbine and compressor maps from the turbocharger manufacturer have been used to calibrate the engine model. It was ensured that all data were satisfying model validity constraints; thus excluding data with low air to fuel ratio or data with slight negative EGR pressure difference etc.

The proposed algorithm has been tested calibrating more than ten of both diesel and gasoline MVM's. Based on the models, multivariate controllers for the air paths have been designed. The MVM structure used was similar to the presented example. The model accuracy was sufficient to get reasonable air path control performance very quickly, at initial iterations of the control development.

8.3 Open Problems

The author of the dissertation thesis suggests to explore the following:

8.3.1 Distribution of the Problem using Dual Decomposition

As shown in Chapter 2, so-called Primal decomposition was used to decompose the primary problem to the distributed problem. Thus, the task becomes tractable even with a large number of data, but the negative effect can be considered the impossibility to apply the constraints to the local and global variables at the same time (i.e. constraints combining both private and complicating variables). This can occur in the model of a Diesel combustion engine in a compressor model, for example. The "living space" of the compressor is limited by the constraints associated with parameters. As was shown in the Chapter 2, the Dual decomposition method does not suffer from this problem, since each sub-problem receives a local copy of a global variable and their equality is required within the master problem constraint. Decomposition of the master problem by means of the Dual decomposition and its testing on the model calibration could be very beneficial for these reasons.

8.3.2 Parallel Computing for Model Calibration

At present, in the distributed calibration, the calculation of sub-problems is solved only serially in the Matlab environment and has not been parallelized. At the same time these subproblems are independent and could be ideally solved in parallel. The time saving could be achieved by distributing the computations, for example, using the Parallel toolbox in the Matlab environment. This has not been done in the presented thesis.

8.4 Fulfillment of the Stated Goals

The fulfillment of the stated goals according to their formulations in Section 1.3 is summarized below.

1. The goal of studying numerical problems that occur during the calibration of nonlinear systems is fulfilled in Chapter 2 and 4. Within the steady-state calibration the common methods are introduced in Chapter 2. It is difficult to find optimal parameters if the initial condition is too far from the optimum, which can be considered as a general disadvantage of these methods. One of the main difficulties of the steady-state calibration is in setting the initial conditions of both states and parameters. The problems are caused especially by the singularities contained in the model if they are not robustly treated. Another major drawback of the steady-state calibration process is associated with a component interconnection where components are forced to receive signals from their predecessors, regardless whether the signal is physical or not. These drawbacks usually lead to a number of numerical problems that are mentioned at the beginning of chapter 4.

2. The goal of analysis of the regularization techniques which can improve a numerical instability of a calibration process is fulfilled in Chapter 4. A solution of the problems mentioned in point 1. includes the introduction of slack-variables that can be viewed as additional degrees of freedom in the model structure. These variables may replace the critical expressions of the component equations, i.e. protect the model singularities. Additional linear constraints are used to enforce the correct range of these variables or signals.

3. The goal of finding a solution of these numerical problems using regularization techniques and development of a methodology for its application is satisfied in Chapters 4-5. As a result, the concept of a regularized component is introduced in Chapter 4. The component has its inputs, outputs and internal variables constrained to limited domains as needed, has a well-defined initial calibration condition and is numerically robust for the calibration process. The methodology for the component regularization is developed and it is accompanied by several examples.

4. The goal of testing a designed solution on a chosen nonlinear model is satisfied in chapters 6-7. Developed methodology is firstly applied on a Mean-

Value model of diesel internal combustion engine and then the model is calibrated using a distributed calibration algorithm. This improved the model's overall fit and increased the overall homogeneity of the results.

References

[1] 4 stroke diesel engine intercooler. https://en.wikipedia.org/wiki/Intercooler#/media/File:Air_Coolers-01.jpg.

[2] Throttle body. https://commons.wikimedia.org/wiki/File:Throttle_body.jpg.

[3] Structure of diesel turbocharger. http://joelcd88.blogspot.cz/2011/07/diesel-engines.html.

[4] E.A. Baskharone. *Principles of Turbomachinery in Air-Breathing Engines*. Cambridge Aerospace Series. Cambridge University Press, 2006. ISBN 9780521858106.

[5] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995. ISBN 9781886529144.

[6] R. Beňo, D. Pachner, and V. Havlena. Robust numerical approach to steady-state calibration of mean-value models. *Control Engineering Practice*, 61:186–197, 2017.

[7] R. Beňo, D. Pachner, and V. Havlena. Robust numerical approach to mean-value modeling of internal combustion engines. *7th IFAC Symposium on Advances in Automotive Control*, Tokyo, Japan, September 2013.

[8] J. Biteus. Mean value model of a heavy duty diesel engine. *Linköpings universitet*, 2004.

[9] D. Björck. *Numerical methods for least squares problems*. SIAM, Philadelphia, 1996.

[10] H.G. Bock and K.J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. *IFAC 9th World Congress*, 9:242–247, 1984.

[11] S. Boyd and M. Johansson. Lectures on distributed optimization. 2010.

[12] S. Boyd, L. Xiao, and A. Mutapcic. Notes on decomposition methods. October 2003. URL https://web.stanford.edu/class/ee392o/decomposition.pdf.

[13] G. De Nicolao, R. Scattolini, and C. Siviero. Modelling the volumetric efficiency of ic engines: Parametric, non-parametric and neural techniques. *Control Engineering Practice*, 4:1405–1415, 1996.

[14] M. Diehl, H.G. Bock, and J.P. Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *Siam J. Control Optim.*, 43(5):1714–1736, 2005.

[15] E. Feru, F. Willems, C. Rojer, B. Jager, and M. Steinbuch. Heat exchanger modeling and identification for control of waste heat recovery systems in diesel engines. *2013 American Control Conference (ACC)*, pages 2860 – 2865, 2013.

[16] R. Fletcher. *Practical methods of optimization*. Number sv. 1 in Wiley-Interscience publication. Wiley, 1987. ISBN 9780471915478.

[17] H.P. Gavin. *The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems*. Department of Civil and Enviromental Engineering, Duke University, 2016.

[18] L. Guzzella and C. Onder. *Introduction to Modeling and Control of Internal Combustion Engine Systems*. Springer, 2009. ISBN 9783642107757.

[19] E. Hairer and G. Wanner. Stiff differential equations solved by radau methods. *Journal of Computational and Applied Mathematics*, 111(1–2):93–111, 1999.

[20] J. Heywood. *Internal Combustion Engine Fundamentals*. Automative technology series. McGraw-Hill Education, 1988. ISBN 9780070286375.

[21] H. Hiereth, K. Drexl, and P. Prenninger. *Charging the Internal Combustion Engine*. Powertrain. Springer, 2007. ISBN 978-32-114711-3-5.

[22] R. Isermann. *Engine Modeling and Control: Modeling and Electronic Management of Internal Combustion Engines*. Springer Berlin Heidelberg, 2014. ISBN 9783642399336.

[23] Mrdjan Jankovic, Miroslava Jankovic, and Ilya Kolmanovsky. Robust nonlinear controller for turbocharged diesel engines. In *American Control Conference, 1998. Proceedings of the 1998*, volume 3, pages 1389–1394. IEEE, 1998.

[24] J.P. Jensen, A.F. Kristensen, S.C. Sorenson, N. Houbak, and E. Hendricks. Mean value modeling of a small turbocharged diesel engine. *International Congress and Exposition, SAE Technical Papers Series*, 1991. doi: 10.4271/910070.

[25] H.K. Khalil. *Nonlinear Systems*. Prentice Hall, 2002. ISBN 9780130673893.

[26] I. Kolmanovsky and A.G. Stefanopoulou. Optimal control techniques for assessing feasibility and defining sub- system level requirements: An automotive case study. *IEEE Transactions on Control Systems Technology*, 9(3): 524–534, 2001.

[27] A. Kozma, C. Savorgnan, and M. Diehl. *Distributed Multiple Shooting for Large Scale Nonlinear Systems*. Springer, 2012. ISBN 9789400770065.

[28] I. Markovsky and S. Van Huffel. Overview of total least squares methods. *Signal Processing*, 87:2283–2302, 2007.

[29] Holger Diedam Pierre-Brice Wieber Moritz Diehl, Hans Georg Bock. Fast direct multiple shooting algorithms for optimal robot control. *Fast Motions in Biomechanics and Robotics*, 2005.

[30] M. Müller. Mean value modeling of turbocharged spark ignition engines. *SAE Technical Papers*, 980784, 1998.

[31] E.Y.K. Ng and N.Y. Liu. *Compressor Instability with Integral Methods*. SpringerLink Engineering. Springer, 2007. ISBN 9783540724124.

[32] K. Nikzadfar and A.H. Shamekhi. An extended mean value model (emvm) for control-oriented modeling of diesel engines transient performance and emissions. *Fuel,*, 154:275–292, 2015. doi: 10.1016/j.fuel.2015.03.070.

[33] D. Pachner, D. Germann, and G. Stewart. Identification techniques for control oriented models of internal combustion engines. *Lecture Notes in Control and Information Sciences*, 418:257–282, 2012.

[34] R.D. Reitz and C.J. Rutland. Development and testing of diesel engine cfd models. *Progress in Energy and Combustion Science*, 21(2):173 – 196, 1995. ISSN 0360-1285.

[35] T. B. Schön, F. Lindsten, J. Dahlin, J. Wågberg, Ch. A. Naesseth, A. Svensson, and L. Dai. Sequential monte carlo methods for system identification. *IFAC-PapersOnLine*, 48(28):775 – 786, 2015. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2015.12.224. 17th IFAC Symposium on System Identification SYSID 2015.

[36] Y. Shi, H. W. Ge, and R. D. Reitz. *Computational Optimization of Internal Combustion Engines*. Springer-Verlag London, 2011. ISBN 978-0-85729-618-4.

[37] E.E. Streit and G.L. Borman. Mathematical simulation of a large turbocharged two-stroke diesel engine. *SAE paper*, (710177), 1971.

[38] M.J. Tenny, S.J. Wright, and J.B. Rawlings. Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming. *Comput. Optim. Appl.*, 28:87–121, 2004.

[39] H. Versteeg. *Introduction to Computational Fluid Dynamics – The Finite Volume Method*. Pearson Higher Education, 2007. ISBN 9780131274983.

[40] J. Wahlström and L. Eriksson. Modelling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 255(7):960–986, 2011.

[41] Y. Wang, X. Tang, and H. He. Thermodynamic simulation of exhaust gas recirculation coolers used on diesels based on finite element method. *7th Intl. Conf. on Sys. Simulation and Scientific Computing*, 7:141–145, 2008.

[42] F.J. Zeleznik and B.J. McBride. *Modeling the Internal Combustion Engine*. NASA - National Aeronautics and Space Administration, 1985.

[43] J. Zhou, L. Fiorentini, M. Canova, and Y.Y. Wang. Coordinated performance optimization of a variable geometry compressor with model predictive control for a turbocharged diesel engine. *IEEE Transactions on Control Systems Technology*, 24:804–816, 2016.

[44] J. Štecha. *Optimal Decision Making and Control*. CTU in Prague, 1999.

Publications of the author

Publications related to the thesis
Journal publications with impact factor

I R. Beňo, D. Pachner and V. Havlena. Robust numerical approach to steady-state calibration of mean-value models. *Control Engineering Practice*, volume 61, pages 186 - 197, April 2017. Co-authorship: 50%, co-autorship of other authors: D. Pachner 40%, V. Havlena 10%.

Publications indexed in Web of Science

II R. Beňo, D. Pachner and V. Havlena. Robust Numerical Approach to Mean-Value Modeling of Internal Combustion Engines, *7th IFAC Symposium on Advances in Automotive Control*, Tokyo, Japan, September 2013. Co-authorship: 50%.

III R. Beňo, D. Pachner, and V. Havlena. Enforcing Stability in Steady-State Optimization, *Sysid 2012, 16th IFAC Symposium on System Identification*, volume 16, July 2012. Co-authorship: 50%.

Publications unrelated to the thesis

Patents

IV R. Beňo and D. Pachner. Compressor Override Control. US 2018/0023490, 2018. Co-authorship: 50%.

Journal publications with impact factor

V D. Valcarcel, A. Neto, I. Carvalho, B. Carvalho, R. Beňo et al. The COMPASS Tokamak Plasma Control Software Performance, *IEEE Transactions on Nuclear Science*, Volume 58, No. 4, August 2011, Co-authorship: 10%.

VI R. Beňo, V. A. Delong, D. Břeň and P. Kulhánek. Notes on the relativistic movement of runaway electrons in parallel electric and magnetic fields. *Physics of Plasmas*, 2016. Co-authorship: 25%.

Peer-reviewed journals with no impact factor

VII R. Beňo and J. John. Modelování a regulace vertikální polohy plazmatu v tokamaku COMPASS (In Czech: Desing and Control of Vertical Plasma Position in Tokamak COMPASS), Automa, Volume 12, Pages 36-40, 2010, Co-authorship: 60%.

VIII R. Beňo and J. John. Modelování zpětnovazebního řízení polohy plazmatu v tokamaku COMPASS (In Czech: Simulation of Feedback Position Control System in Tokamak COMPASS), *Czechoslovak Journal of Physics*, Volume 59, No. 4, Pages 242-245, 2009, Co-authorship: 60%.

Other publications - Books

IX Petr Kulhánek a kol.: Astronomie a fyzika - Svítání (In Czech: Petr Kulhánek et al.: Astronomy and Physics - The Sunrise), *AGA - Aldebaran Group for Astrophysics*, ISBN: 978-80-904582-6-0, 2014. Co-authorship: 5%.

Other publications - Popular-educational articles

X Radek Beňo. Králík, který běhá po Měsíci (In Czech: A rabbit is running across the Moon), *Aldebaran Bulletin*, vol. 40, 2013, online: http://www.aldebaran.cz/bulletin/2013_40_kra.php

XI Radek Beňo. Superoko Gaia (In Czech: Supereye Gaia), *Aldebaran Bulletin*, vol. 2, 2014, online: http://www.aldebaran.cz/bulletin/2014_02_gai.php

XII Radek Beňo. Rande s kometou (In Czech: A date with a comet), *Aldebaran Bulletin*, vol. 29, 2014, online: http://www.aldebaran.cz/bulletin/2014_29_ros.php

XIII Radek Beňo. Udělení Nobelovy ceny za fyziku pro rok 2014 (In Czech: The Nobel Prize Award for the year 2014), *Aldebaran Bulletin*, vol. 33, 2014, online: http://www.aldebaran.cz/bulletin/2014_33_nob.php

XIV Radek Beňo. Indové na Marsu (In Czech: Indos on the Mars), *Aldebaran Bulletin*, vol. 36, 2014, online: http://www.aldebaran.cz/bulletin/2014_36_rad.php

XV Radek Beňo. Vítejte na kometě (In Czech: Welcome on the comet), *Aldebaran Bulletin*, vol. 37, 2014, online: http://www.aldebaran.cz/bulletin/2014_37_rad.php

XVI Radek Beňo. Shledání s modulem Beagle 2 (In Czech: Reunion with the module Beagle 2), *Aldebaran Bulletin*, vol. 9, 2015, online: http://www.aldebaran.cz/bulletin/2015_09_bra.php

XVII Radek Beňo. Philae volat domů (In Czech: Philae phone home), *Aldebaran Bulletin*, vol. 24, 2015, online: http://www.aldebaran.cz/bulletin/2015_24_kom.php

XVIII Radek Beňo. Indická vesmírná observatoř Astrosat (In Czech: Indian space observatory Astrosat), *Aldebaran Bulletin*, vol. 34, 2015, online: http://www.aldebaran.cz/bulletin/2015 _34_ind.php

XIX Radek Beňo. Nám, nám narodil se Vendelín (In Czech: The birth of Vendelin), *Aldebaran Bulletin*, vol. 3, 2016, online: http://aldebaran.cz/bulletin/ 2016_03_ven.php

XX Radek Beňo. Stephen Hawking: Breakthrough Starshot, *Aldebaran Bulletin*, vol. 16, 2016, online: http://aldebaran.cz/bulletin/2016_16_pla.php

XXI Radek Beňo. Dotek roboreality (In Czech: Touch of the roboreality), *Aldebaran Bulletin*, vol. 17, 2016, online: http://aldebaran.cz/bulletin/2016_17_rad.php

XXII Radek Beňo. Rosetta - mise splněna (In Czech: Rosetta - mission accomplished), *Aldebaran Bulletin*, vol. 38, 2016, online: http://aldebaran.cz/bulletin/2016_38_ros.php

Curriculum Vitae of the Author

Radek Beňo was born in Zlín, Czech Republic, in 1985. He received his bachelor degree in Electrical Engineering from Czech Technical University (CTU) in Prague, the Czech Republic in 2009 and his master degree in Electrical Engineering and Cybernetics at the same university in 2011. At the same year, 2011, he started his Ph.D. studies in the Control Engineering and Robotics at CTU Prague. The main topic of the research is the distributed identification of nonlinear systems.

Radek Beňo is co-author of international conference paper in the area of modelling and control (IFAC Symposium on System Identification – 2012) and automotive applications of advanced control methods (IFAC Symposium on Advances in Automotive Control – 2013). He is a holder of the US patent in the domain of applications of advanced control methods in the automotive.

Radek is also interested in plasma physics and tokamak control. He is co-author of several papers in national journals (Czechoslovak Journal of Physics – 2009 and Automa – 2010) and co-author of several papers in international journals (IEEE Transactions on Nuclear Science – 2011 and Physics of Plasmas – 2016). Within the Aldebaran group, he is engaged in the popularization of astrophysics.

From 2007 to 2011 was Radek employed as a technician worker at Institute of Plasma Physics AS CR in Prague, where he was a member of a feedback group. His work was focused on modelling, simulation and feedback control of plasma position in tokamak.

From 2012 to 2016 was Radek employed as a a Research Engineer in Honeywell Prague Laboratory in Prague. His work was focused on developing new algorithms, specially for internal combustion diesel engine calibration.

Prague, February 2018

Radek Beňo