Master of Science Thesis in Mechanical and Electrical Engineering
Department of Electrical Engineering, Linköping University, 2022

Modelling and Optimal Control of a Variable Nozzle Turbine in an SI Engine for Maximum Performance

Karl Bolin and Emil Fransson Brunberg

LINKÖPING UNIVERSITY

Master of Science Thesis in Mechanical and Electrical Engineering

Modelling and Optimal Control of a Variable Nozzle Turbine in an SI Engine for Maximum Performance

Karl Bolin and Emil Fransson Brunberg

LiTH-ISY-EX--22/5510--SE

Supervisor: **Oskar Lind Jonsson**
ISY, Linköpings universitet
Bohan Liang
Aurobay
Gustav Mann
Aurobay

Examiner: **Lars Eriksson**
ISY, Linköpings universitet

Vehicular Systems
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

The ever increasing demands on today's engine performance and emissions control is forcing the automotive industry to make use of innovative solutions. One of these is to apply the technology of VNT turbos on commercial petrol vehicles. When using a VNT turbo, the aspect ratio of the turbine can be altered while driving to suit the current operating window. In order to actually gain performance while using a VNT, the vanes have to be properly controlled using a suitable control strategy. In this project, direct collocation have been utilized through the usage of YOP which is an adaptation of CasADi in MATLAB to solve non-linear optimization problems. Comprehensive models of the turbocharger and the cylinders have been built and validated to properly represent a VEP4 LP engine from AUROBAY. The models are implemented in YOP to create and simulate different OCPs using the turbo speed as state and position of the vanes as control signal. With this model in YOP together with the air mass flow $\dot{m}_{air}$ as reference, a good reference following together with decent values for relevant parameters can be accomplished. Other objective functions such as minimum time and maximal volumetric efficiency are also investigated in the project which yield likewise results. From the results it can be concluded that this type of model and control strategy can be used with success when studying optimal control of a VNT turbo.

Acknowledgments

To begin with, we are grateful for the opportunity to study this idea presented by AUROBAY, supervised by Bohan Liang and Gustav Mann in cooperation with Fredrik Wemmert. They have assisted with insightful support, guidance and positive encouragement in the project as well as in life.

We would like to thank Oscar Lind Jonsson at Linköping University for accepting us to be his first students to supervise as a new doctoral student. It has been a delight to discuss the project while having a laugh at the r/formuladank subreddit

We would also like to give a special thanks to Viktor Leek at Scania for helping us with all types of questions regarding YOP which turned out to be vital for the thesis.

Last but not least, the man who coined the expression of "*förenkla hänsynslöst*", Lars Eriksson at Linköping University, our examiner for the project, is worthy of a special thanks. A bottomless fountain of knowledge within the fields.

Linköping, June 2022
Karl Bolin and Emil Fransson Brunberg

Contents

List of Figures

Notation

Acronyms

Notation	Description
VNT	Variable Nozzle Turbine
VGT	Variable Geometry Turbine
FGT	Fixed Geometry Turbine
E-turbo	Electric Turbocharger
AR	Aspect Ratio
COM	Control-Oriented Model
MAP	Manifold Air Pressure
LPV	Linear Parameter Varying
MPC	Model Predictive Control
EGR	Exhaust Gas Recirculation
x_{egr}	EGR-fraction
PID	Proportional, Integral, Differential (Controller)
CI	Compression Ignition
SI	Spark Ignition
COM	Control-oriented Model
MVM	Mean Value Model
MVEM	Mean Value Engine Model
ECU	Electronic Control Unit
BSR	Blade Speed Ratio
LQR	Linear Quadratic Regulator
NLP	Nonlinear Programming
OCP	Optimal Control Problem
LP-filter	Low-Pass Filter
GPF	Gasoline Particle Filter

Symbols

Notation	Description	Unit
$\dot{m}_{air}$	Air Mass Flow	kg/s
$\dot{m}_{exh}$	Exhaust Mass Flow	kg/s
$\dot{m}_c$	Compressor Mass Flow	kg/s
$\dot{m}_{c,corr}$	Corrected Compressor Mass Flow	kg/s
m_{stroke}	Air Mass Flow per stroke	mg/stroke
ρ_{air}	Air Density	kg/m^3
N_{eng}	Engine Speed	rpm
N_c	Compressor Speed	rpm
$N_{c,corr}$	Corrected Compressor Speed	rpm
N_t	Turbine Speed	rpm
N_{turbo}	Turbocharger Speed	rpm
$\dot{N}_{turbo}$	Turbo Speed Derivative	rpm/s
M_{fric}	Friction uncertainty Torque	Nm
T_1	Intake Temperature	K
T_2	Compressor Outlet Temperature	K
T_3	Turbine Inlet Temperature	K
T_4	Exhaust Temperature	K
$T_{GPF,ds}$	GPF Downstream Temperature	K
p_{amb}	Ambient Pressure	Pa
p_1	Intake Pressure	Pa
p_2	Boost Pressure	Pa
p_3	Turbine Inlet Pressure	Pa
p_4	Exhaust Pressure	Pa
$p_{GPF,ds}$	GPF Downstream Pressure	Pa
$p_{GPF,us}$	GPF Upstream Pressure	Pa
Π_c	Compressor Pressure Ratio	-
p_r	Turbine Pressure Ratio	-
Π	Turbine Pressure Ratio Function	-
Ψ	Choke Function	-
η_c	Compressor Efficiency	-
$\eta_{c,obs}$	Observed Compressor Efficiency	-
η_t	Turbine Efficiency	-
η_m	Mechanical Efficiency	-
η_{vol}	Volumetric Efficiency	-
$\dot{W}_{c,ideal}$	Ideal Compressor Power	W
$\dot{W}_c$	Compressor Power	W
$\dot{W}_{t,ideal}$	Ideal Turbine Power	W
$\dot{W}_t$	Turbine Power	W
$A_{eff,rear}$	Rear Effective Area of GPF	m^2
$A_{eff,front}$	Front Effective Area of GPF	m^2
A_{eff}	Turbine Effetive Area	m^2
X_{vnt}	VNT Vanes Position	-
λ	Air-to-fuel Equivalence Ratio	-
λ_o	Oxygen-fuel Equivalence Ratio	-
BSR	Blade Speed Ratio	-
TSP	Turbine Speed Parameter	$rpm/\sqrt{K}$
BSFC	Brake-specific Fuel Consuption	g/kWh

Constants

Notation	Description	Value	Unit
n_r	Stroke-Revolution Ratio	2	-
M	Air Molar Mass	29	g/mol
V_D	Total Cylinder Volume	$1.969 \cdot 10^{-3}$	m^3
T_s	Sample Time	0.01	s
T_{ref}	Reference Temperature	298.15	K
$K_{inertia}$	Intertia Derivative Gain	5760000	-
$c_{p,air}$	Intake Specific Heat Capacity	1062.676	J/(kg · K)
$c_{p,exh}$	Exhaust Specific Heat Capacity	1183.875	J/(kg · K)
p_{ref}	Reference Pressure	101530	Pa
AFR	Air-to-fuel Ratio	14.7	-
γ_{air}	Intake Specific Heat Ratio	1.37	-
γ_{exh}	Exhaust Specific Heat Ratio	1.32	-
$R_{air,mol}$	Intake Air Ideal Gas Constant	8314.5	J/(kg mol · K)
R_{air}	Intake Air Ideal Gas Constant	286.7069	J/(kg · K)
R_{exh}	Exhaust Air Ideal Gas Constant	280	J/(kg · K)
x_n	Turbine Inlet Temperature Polynom Constants	-	-

1 Introduction

In this chapter, the purpose and background to the thesis work are presented together with previous research within the field in order to give a comprehensive foundation of the project.

1.1 Purpose and Goal

The goal of this thesis is to develop and validate models for a variable nozzle turbine (VNT) and components affected by its dynamics for a spark ignited (SI) engine. Compared to a traditional turbocharger, a VNT turbo can provide an engine with faster response at a wider range of engine speeds as well as increasing the efficiency of the engine, and therefore also reducing the emissions. To accomplish this, a control strategy will be developed that controls the position of the inlet vanes to the turbine for optimised performance.

The purpose of this thesis is to help AUROBAY get an edge in the automotive engine market by contributing to their research of VNT turbochargers and allowing AUROBAY to build more powerful and less fuel consuming engines. This report will also help combustion engines fulfill the ever increasing emission regulations on the automotive industry.

1.2 Previous Research within the Field

A lot of research have been conducted within the field of turbochargers in combustion engines. Below, some of the research of VNT and how to model and control them is briefly presented.

1.2.1 Turbocharger Configurations

To power boost an engine, the technology of a variable geometry turbine (VGT) or VNT is applied in a wide range today. One reason for this is to fulfill the harder emission regulations on the engines since it allows downsizing of an engine, without losing too much performance and torque characteristics [1]. This was to some extent solved by using a fixed geometry turbocharger (FGT), but these suffer from dilemmas as slow transient response (turbo-lag), a fixed aspect ratio (AR) for an optimised specific operation-region, and it regulates the boost-pressure using a waste-gate to satisfy the needed amount of air into the cylinder to fulfill the load-demand. A solution to the turbo-lag was to make use of sequential turbocharging, where at least two turbochargers of different size operating complete or partially in sequence. At low engine speeds a smaller turbocharger with smaller inertia is being used, and then a transition to a larger turbocharger with larger inertia would occur for intermediate and high engine speeds. But by doing this mainly extra weight and thermal inertia disadvantages occur [2]. Another solution is to use an electrified turbocharger (E-turbo), where a battery and an electric motor is connected to the turbocharger, where the electric motor can spool up the turbine or act as a generator to charge the battery depending on the current air-flow and desired torque from the driver. However, these E-turbos add more complexity to an already complex system, [3], as well as more cost [2].

By instead making use of a VNT where the angle of the vanes in front of the turbine-blades can be controlled, the operation range of where the performance of the engine is enhanced increases and a more precise control of the boost-pressure is possible resulting in a cleaner and more efficient engine. Also a decrease in pumping losses at higher engine speeds occurs [2, 4]. VNT devices comes in different designs but the main purpose is to alter the cross sectional area of the inlet to the turbine which guides exhaust gas into the turbine rotor. By doing this, the AR is no longer a fixed value as it is for the FGT and an increase in kinetic energy of the exhaust gas can be achieved for the lower engine speeds and the turbine can then easily be spooled up. Since there is no waste-gate to handle excess exhaust gases, the vanes are controlled to open up to reduce boost pressure and prevent over-boosting. By other means, a VNT can alter the flow conditions upstream of the turbine without having to alter the moment of inertia. Worth noticing is that the peak efficiency for a VNT is generally lower than for a FGT due to leakage in the turbine casing and the moving parts, but the overall efficiency is greater for the VNT thanks to the wider operation range. There are different mechanical variants of VNT. [2] states the two main types turbines to be axial and radial turbines. What differs between them are at what angle the exhaust gas enters the rotor blades, where radial turbine means entering perpendicular to the blades and are then redirected 90 degrees by the rotor before exiting the housing in the axial direction. An axial turbine works in the opposite manner where the exhaust gas enter axially and exits radially. In addition to this, different configurations are presented that can be made using either a sliding nozzle or pivoting vanes to manipulate the AR. [5] state that the use of VNT can improve the boost capability of the turbo in both low-end torque and high-speed operations where a waste-gate

instead is limited in the power such a system can create. Due to improved efficiency of the VNT compared to traditional turbos, the fuel consumption could decrease at the same time as the performance of the engine increases.

The turbocharger and the exhaust gas re-circulation (EGR) become the coupling components between the intake and exhaust manifolds that are able to control the airpath and this could be considered as a multiple input multiple output (MIMO) control problem if one wanted to control them simultaneously. Instead [6] presents a control scheme where the VNT is decoupled and considered a single input single output (SISO) control problem. To control the turbocharger in real time is a subtle task and thanks to further research within the field and improving hardware and software, there are different ways of creating and applying suitable control strategies.

1.2.2 Modelling

It is usual to strive to make use of a control-oriented model to later be able to combine and implement a control design. A suggestion to create a control oriented model is to make use of a mean value engine model (MVEM) that capture the dynamics of the system and non-linear effects of the gas-flow by using differential equations. The usage of MVEM is wide and considered to be a good tool for utilising engine management, control and supervision [7]. And in [8] Eriksson and Wahlström develops a component based MVEM of a diesel engine using VNT and EGR to capture non-linear system dynamics and after a validation states that the model captures system properties that are important for successful engine control, such as non-minimum phase behaviour and couplings between channels. It is also emphasised to use as few states as possible to minimise computational cost and to look both at the submodels and complete model when optimising the parameters. In the conclusion it is stated that the mean relative error after both static and dynamic tests resulted in 5.8 % or lower. An extension to this is the mean value models (MVM) presented in [9] where the control-oritented model is created (COM) for a turbine that is modeled with an actively controlled turbocharger (ACT). What differs a MVM to a COM is that a COM is equipped with an additional feature to calculate the turbine angular velocity and torque to be able to determine the actual power. The similarities between simulation and experimental results were good, but it is stated that the model is incapable to replicate the more complex phenomena as hysteresis and choking.

Another easy and standard method to model the turbo is to set the mechanical efficiency of the turbo to be a fixed constant but this may lead to modelling errors from over/under predicted compressor power. So instead it is common to make use of manufacturer maps and look-up tables for the efficiency and use map-based models of the turbine and compressor. However, here the issues are that the maps are most suited for steady-state operation and does not cover all the operation points for transients. Furthermore it becomes a 3D map when the VNT position is included and thus added complexity, heavy calculations and problems when interpolating and extrapolating between or outside the predefined

data points of maps [10]. Gonzales et al. [11] suggest to make use of maps but presents different methods to reduce the compressor and turbine maps in to a set of equations by defining a series of non-dimensional and normalised variables that define a plane transformation. In the same paper, additional variables and an optimisation process that would make maps more accurate as well as less computationally heavy than the original maps are presented. In [12], the model of the turbo as a polytropic process with zero and one dimensional gas dynamic code is designed, capable of interpolating and extrapolating to make use of the manufacturer maps. In [13] it is stated that the intake manifold air pressure (MAP) and intake manifold air flow (MAF) are the parameters mainly responsible for the transient response, and therefore model the airpath using a linear parameter varying (LPV) model. An LPV-model is utilised to develop open-loop systems. Further, H. Das et al. also model the mass flow through the turbine using the standard orifice flow equation.

T. Zeng et al. [10] claims that the efficiency-map from manufacturer does not provide full coverage of the operating points and need to interpolate and extrapolate using polynomials as a function of the blade speed ratio (BSR), which could lead to large model errors. Instead, a physics-based turbine model based on turbine downstream conditions using the Euler turbine equation and applying a friction-loss model is suggested, where the efficiency is modeled as a function of vane position and turbine shaft speed. This results in a 10.1% transient error compared to 22.8% using the map-based model.

L. Eriksson et al. in [8] model a MVM of a cylinder is, where the parameter values are achieved from making use of least-squares optimisation. The cylinder model is built by three sub-models describing the mass flow for the gas and fuel entering and leaving the cylinder, the exhaust manifold temperature and the engine torque. In the same paper, the volumetric efficiency is modelled as a function of engine speed, intake manifold pressure and three tuning parameters. [14, 15] comes to the conclusion that the volumetric efficiency is determined by the intake manifold pressure and by varying the duration of opening time for the inlet valve/amount of new air allowed into the cylinder depending on the engine speed.

1.2.3 Control Theory

To get the desired performance out of the engine, various control methods can be used to control position of the VNT vanes. One rather old and simple method developed for an CI engine is to have the vanes position as a function of engine speed and load in Brake Mean Effective Pressure (BMEP) [16]. Controlling the VNT during a transient is however very complex due to the nonlinear characteristics of the VNT and how it affects the very important volumetric efficiency of an SI engine, and therefore it requires a more advanced control strategy. A semi-physical feed-forward control based on predictions of possible VNT trajectories is a good example of a complex but well-performing control strategy [17]. Nielsen et al. have researched EGR-VGT control and use a PID controller with

oxygen/fuel ratio λ_o, instead of the classic air/fuel ratio λ, and EGR-fraction x_{egr} as feedback and performance variables to significantly reduce the pumping losses in a CI engine. The variable x_{egr} is here controlled by the position of the VNT [18]. LQR is a control strategy that can achieve good results in terms of optimisation by controlling the VNT position. However, even faster dynamic response can be achieved by combining the VNT control with, for example, control of the effective area of the EGR [19]. Wahlström explains that a VNT in a CI engine has nonlinear characteristics and therefore investigates an MPC to control the VNT and EGR and compares it with a PID controller which is not as good with nonlinear systems but considerably easier. The MPC gives a faster and more optimal control but also requires heavier data calculations. The most notorious reason for this is that the developed PID controller assumes there is no cross-coupling in the system [20]. The heavier data calculations is however getting easier to handle with the ever-increasing computational power [21]. A fuzzy adaptive control have the potential to increase performance compared to a PID, but without increasing the required computational power as much as an MPC [22]. Linearised MPC (LMPC) is also a good option to control the VNT. In addition to the great control of the turbine, MPC can also easily handle the problem of turbocharger overspeed, i.e. instability, by adding it as a constraint [23].

To solve trajectory optimisation problems, a transcription method can be used to instead create a constrained parameter optimisation problem and utilise nonlinear programming (NLP). There are different ways to transcript the problem. Direct single and multiple shooting are two options to solve optimal trajectory problems, approximating the trajectories using simulations. These methods works best when there is few path constraints and a simple control. Another solution is to use collocation methods which are based on functions instead of simulations and are better suited for more complicated control and in cases where there are more path constraints. There are different types of collocation methods, with different number of trajectory segments and order of the polynomials used [24]. Pseudospectral collocation methods can be used to solve optimal control problems and can represent state trajectories as high order polynomials. For a time optimal transient in a heavy duty CI engine with constant speed, while controlling the VNT, no EGR fraction constraints and a specific minimum λ_0-value, the VNT should close gradually to not increase the back pressure and temperature too much. When the transient is over the VNT should open up for the engine to achieve the desired torque level. The EGR meanwhile, is closed during the entire transient. For the same case but with a fuel optimal transient, both actuators should be closed. The VNT should open when the intake pressure is high enough for the transient end conditions [25]. A state feedback controller for the boost pressure with VNT control is an additional alternative. Although performance is comparable with current map-based solutions, the number of variables is reduced and the controller is therefore easier to understand. Furthermore, [21] suggests that a more complex algorithm, such as MPC, is required to catch the nonlinearities in the system. Existing engine models can be combined with a neural networks controller, this can achieve very good results in CI engines and it

is relatively simple to add further inputs and outputs, i.e. components, into the system [26].

1.3 Problem Formulation

The usage of a VNT in an SI engine is introducing three main challenges for the engine that needs to be solved in order for the VNT to be effective. These are as follows:

- VNT turbo challenge 1: exhaust flow

The power generated by the turbine can be expressed as:

$$\dot{W}_{t,ideal} = \dot{m}_{exh} c_{p,exh} T_3 \left(1 - \left(\frac{p4}{p3} \right)^{\frac{\gamma_{exh}-1}{\gamma_{exh}}} \right) \tag{1.1}$$

By closing the VNT rack, the turbine inlet pressure p_3 increases and the exhaust mass flow $\dot{m}_{exh}$ decreases. If the exhaust mass flow decreases too much, it will result in a net loss of power for the compressor.

- VNT turbo challenge 2: cylinder air flow and pumping losses

The increased turbine inlet pressure p_3 also makes it harder for the exhaust gas to exit the combustion chamber. This in turn reduces the volumetric efficiency of the cylinders. In order to keep air flowing through the cylinders, a higher boost is needed. Furthermore, also the pumping losses will increase with increased exhaust pressure.

- VNT turbo challenge 3: stability

The usage of a VNT instead of a wastegate makes the system a feedback system since all of the air going in through the compressor also goes through the cylinders and turbine. In the case of a wastegate on a FGT, excess air that is not wanted to charge the turbine can pass past the turbine and prevent further boost pressure build-up. For a VNT, this will have to be controlled through the VNT position. If this is not properly done, an increase in boost pressure will lead to increased cylinder air flow, which leads to increased exhaust gas energy, which leads to increased turbine power, which in turn leads to an even higher increase in boost pressure. In other words, the system becomes unstable. If the air flow into the engine is higher than demanded, the engine will increase the amount of injected fuel in order to keep a correct AFR, this will increase the boost pressure and add to the instability problem. If the load is higher than expected, the timing of the ignition can be delayed, this decreases the efficiency of the engine, which gives warmer exhaust gases and therefore increases the power output of the turbine which also contributes to the instability problem.

From these challenges, the following questions can be formulated and answered later in the report.

- How should the turbocharger and cylinders be modelled to properly be able to analyse the parameters from the challenges?
- How to build a control strategy based on a non-linear system?
- How to optimally control the actuator of the VNT for different transients to avoid the mentioned challenges?

1.4 Stopping Criterion

The project will be considered done when a VNT turbo have been modelled that can calculate the volumetric efficiency and created turbine power, a cylinder flow model have been developed that can handle large pressure variations between intake and exhaust manifold and an exhaust temperature model have been developed. These models can be integrated into each other and give good calculations compared to experimental data. The models are able to give optimal performance in terms of transient properties by controlling the VNT vanes position. The results are illustrated by e.g. plotting volumetric efficiency, turbine power and VNT vanes position over a transient for the developed optimal control strategy and compare it to sub-optimal control strategies and an ordinary FGT.

The project will be considered a success if the controller described in this report can be directly implemented in current and future engines of AUROBAY and give the same performance advantages as described in the conclusion of this report. The report should also be useful for future research conducted by AUROBAY and help them strengthen their position in the Automotive engine market.

1.5 Delimitations

This thesis work will primary focus on modelling and control of the VNT turbocharger and the cylinder flow model, meaning that other components in the engine, e.g. EGR control, are either already manifested and provided by AUROBAY or neglected. The intake pressure for calculations and simulations will always be at or higher than the ambient atmospheric pressure. If a lower pressure is desired, perfect throttle control will be assumed. Experimental data will only be extracted from test performed on the provided engine platform of a VEP4 LP-Miller engine. VEP4 stands for Volvo Engine Architecture petrol with four cylinders in an inline configuration, LP for low power and Miller is the type of thermal dynamic cycle that is used in the four-stroke combustion.

1.6 Report Outline

In the first chapter, the problem is formulated, given a background and related to previous work within the engine modelling and control field. In the second chapter, the theory behind an internal combustion engine (ICE) is explained as well as relevant control strategies. Chapter three describes the experiments done for

data collection to build the models and controller. Chapter four will describe the models and how they have been built. Chapter five describes the control strategy and how it has been implemented into the system. Chapter six contains the final results and validations of the work. In chapter seven the results are discussed regarding to limitations and simplifications. In chapter eight, proposals of future work to improve the models and controllers are presented. Finally, in chapter nine, the conclusion of the master thesis is presented.

1.7 Contributions

Despite the fact that both students have different master profiles the project have not given the possibility to properly split the work between these profiles. This is because the entire modelling and mechanical engineering part of the project were needed to be finished before the control strategy and electrical engineering part could get started. Because of this, a lot of the work have been carried out by both students together. The literature studies of each subject however have been done by the respective student.

2

Theory of Internal Combustion Engines, Models and Control Strategies

In this chapter, all of the theory needed in order to understand the models and controllers will be explained.

2.1 The Combustion Cycle

As previously stated, this report is based on a four-cylinder four-stroke SI engine. The components that will be studied in detail are the turbocharger and the cylinder as these parts are the most relevant to the project. Since the volume of air between the physical components is very small in the current engine compared to other, no control volumes will be used for modelling and calculations. See Figure 2.1 [27] for an illustration of where index 1,2,3 and 4 are located.

2.1.1 Static Compressor

The compressor is one of the two main parts of a turbocharger. It is mechanically coupled to the turbine with a solid shaft and is driven by the energy in the warm exhaust gases. The compressor is used to increase the intake manifold pressure (boost pressure) by compressing the air going into the intake manifold, and thus increasing the possible amount of air into the cylinder allowing more power to be produced by the engine. Normalised corrected quantities for the mass flow and speed of the compressor can be used to make sure different surrounding conditions, e.g. hot or cold temperature, high or low altitude, are being taken into account when executing calculations [27]. Important quantities to estimate are how the boost pressure p_2 increases are the compressor speed N_c, mass flow through the compressor $\dot{m}_{air}$, the intake pressure p_1, intake temperature T_1 and the compressor power $\dot{W}_c$.

2.1.2 Intake Manifold

The intake manifold is filled with air passing through the throttle body and the compressor from the air intake, and guides the air into the cylinders. A larger intake volume allows for more air being available for the cylinders to use. Parameters representing the intake manifold are the intake manifold temperature T_2 and boost pressure p_2.

2.1.3 Cylinder

It is in the cylinder that the combustion takes place. In an SI four-stroke cycle the cylinder first fills up with fresh air during the intake stroke, then compresses it during the compression stroke, the spark plug then ignites and the gas explodes at top-dead-centre and the piston gets pushed down by the expansion during the power stroke, and finally when all work from the power stroke have been carried out at bottom-dead-centre the hot exhausts gets pushed out during the exhaust stroke. The specific effects of valve overlap will not be taken into consideration in this work. At steady state the air mass entering into the intake manifold is equal to the air mass into the cylinder. Important quantities representing the cylinder is the boost pressure p_2, and turbine inlet pressure p_3, cylinder mass flow $\dot{m}_{air}$ and volumetric efficiency η_{vol}.

2.1.4 Exhaust Manifold

The exhaust gas gets pushed out of the cylinder and into the exhaust manifold that guides the gas to the turbine. Parameters representing the exhaust manifold are the exhaust temperature T_3 and exhaust manifold pressure p_3.

2.1.5 Static Turbine

The turbine is driven by the warm exhaust gases from the cylinders and in turn drives the compressor. In other words, the turbine is responsible for recovering some of the lost energy stored in the exhaust gases and use it to drive the compressor. Important quantities representing the turbine is the turbine speed N_t, the turbine power $\dot{W}_t$, the turbine mass flow $\dot{m}_t$, the effective area A_{eff} and the position of the vanes X_{vnt}.

2.1.6 Exhaust System

In the exhaust system the after-treatment of exhaust gases takes place. It consists of a catalyst and a gasoline particle filter. Relevant parameters are the exhaust temperature T_4, exhaust pressure p_4 and the front and rear effective area of the GPF, $A_{eff,rear}$ and $A_{eff,front}$.

2.1.7 Actuator

The actuators are the construction that puts the mechanical parts (in this case pivoting vanes) in motion by converting energy and reading signals received from the control strategy calculated in the electronic control unit (ECU). The actuator is assumed to have no dynamics in this project and move exactly as the control signal does.

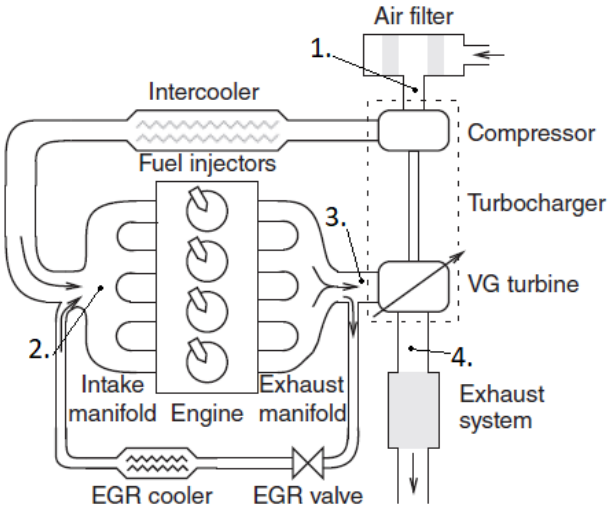


Figure 2.1: *Sketch showing the engine configuration of a direct injected SI VNT engine.*

2.2 Modelling

The engine is modelled using MVEM. MVEM is a control oriented designing method to study the air path of the engine, and is able to describe variations that are slower than an engine cycle [28]. Using MVEM the cylinders are not modeled individually but instead a mean value for one or several cycles are being used [29]. The volumes of the inlet and exhaust manifold are assumed to be relatively small and thus control volumes between the turbocharger and the cylinder are neglected when modelling. Both the cylinder mass flow and the turbo-shaft connects the compressor and turbine making this a natural feedback system since no mass is leaving the system.

2.2.1 Static and Dynamic Models

Due to the neglecting the control volumes the relationship among the compressor quantities is assumed to be static. The same goes for the turbine quantities. A dynamic model can be defined by a differential equation, where the output (k)

depends on the previously simulated output value (k-1). At steady state the rate of change for the next value is zero.

2.2.2 States

When modelling MVEM states are commonly used to express mass and energy in measurable quantities such as temperature and pressure. Control volumes are often used as an intermediate step between two components that can express the quantities using representative differential equations.

2.2.3 Incompressible Flow Restrictions

For flows with velocities up to Mach 0.2-0.3, the air can be modeled as incompressible which is the case for most flows in the engine. This is due to the small pressure variation. If the velocity is higher it can be modeled as compressible as the pressure variation increases. The incompressible flow can be divided into laminar and turbulent flow [27].

2.2.4 Maps

A 3D map can be built using x,y and z values, creating a surface. This is useful when building a map-based model where x and y as inputs generate z as output either by interpolating or extrapolating. A 2D map, or a normal table, can be built likewise, then only using x as input and y as the output.

2.2.5 Black Box and Grey Box

In modelling, a black box is seen as a system where the known input generates an output without any knowledge of how the output is calculated internally [30]. A grey box is very much like a black box, but with the user having some knowledge of the relation between the input and output instead of no knowledge at all. How the input affects the output in a grey box can be somewhat explained by a theoretical basis [31].

2.2.6 Low-pass Filter

Low-pass filter is a filter that passes signals with a frequency lower than a specific value. It is beneficial to use an LP-filter to reduce high oscillations in a signal but with the trade-off that the signal could get a time delay depending on how strict the filter is. The filter is defined as a discrete transfer function in the SIMULINK model.

2.3 Control Strategies

Control strategies are used in order to achieve as good performance output as possible of a system while fulfilling different constraints and goals. To achieve this

there is an actuator present to be controlled with signals from a regulator which affects the system. A system often contains one or several performance outputs z, measured outputs y, control signals u and disturbances w [27]. According to Max Johansson (2022) from Linköping University, to solve an optimal control problem (OCP), the objective function should preferably be convex to know if the solution is a global optimum. Below is a standard formulation of an OCP:

$$\begin{aligned} \min_{u(t)} \quad & E(T, x(T), P) + \int_0^T L(t, x(t), u(t), P)\, dt \\ \text{s.t.} \quad & x(0) = 0, \\ & \dot{x}(t) = f(t, x(t), u(t), P) && t \in [0, T] \\ & h(t, x(t), u(t), P), \leq 0 && t \in [0, T] \\ & h_T(T, x(T), u(T), P) \leq 0 \end{aligned} \tag{2.1}$$

where t is the independent time variable, x is the state, P is the free optimisation parameter vector and T is the final time value. $\dot{x}(t)$ is the state derivative with respect to the time variable and x_0 is the state initial value. $f(t, x(t), u(t), P)$ is the right-hand side of the ordinary differential equation (ODE), $h(t, x(t), u(t), P)$ and $h_T(T, x(T), u(T), P)$ are the inequality path constraints and terminal inequality constraints respectively. The term E is the terminal state cost, also known as the Mayer term. L is the Lagrange term, also known as the integral cost function. When the objective function is a combination of both of these terms, the objective function can be called a Bolza objective [29].

The OCP will be converted to an NLP when using direct methods of optimisation to solve the problem, which will be discussed later. A basic formulation of an NLP is as follows.

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & \underline{x} \leq x \leq \overline{x} \\ & h(x) \leq 0 \\ & g(x) = 0 \end{aligned} \tag{2.2}$$

where $f(x)$ is the objective function, $\underline{x}$ is the lower bound value, $\overline{x}$ is the upper bound value, $h(x)$ inequality constraints and $g(x)$ the equality constraints.

2.3.1 Regulator

The regulator will control the position of the vanes of the VNT with an actuator in order to control the airflow through the cylinders. In this thesis, PID and direct collocation will be utilised and compared as the control strategies in order to optimise transient demands.

2.3.2 Direct Single and Multiple Shooting

Direct single shooting is done by discretising the continuous time dynamics with numerical integration of the state trajectory. In single shooting, the integration is done from start to finish in one single "shot". This method gives poor performance if the system is unstable and instead multiple shooting can be used which can handle this problem [29, 32]. As the name suggest, multiple "shots" are shot over the time interval instead of just one and is a suitable solution for simple control problems with no path constraints. To convert the OCP to an NLP, the independent variable is discretised into a finite number of intervals. For each interval, the control signal is kept constant making it piecewise constant seen over the entire time period. To keep the state trajectory continuous, a constraint is added to the NLP to make sure the trajectories are together between the intervals. Fixed step explicit Runge-Kutta is a common integration method for multiple shooting. More in-depth equations for single and multiple shooting can be found in Viktor Leek's thesis for development of YOP [29].

2.3.3 Direct Collocation

As for multiple shooting, direct collocation is primarily used to transcript an OCP into a NLP, creating a system of algebraic equations. However, collocation is more suitable for problems with complicated control and path-constraints. The main idea is to approximate the state and control using polynomials. In addition, the direct collocation make use of an implicit integrator as Runge-Kutta [24]. By choosing direct collocation instead of direct multiple shooting a trade in making the problem larger but with a decreased non-linearity is made. By transcripting the problem to a NLP the control signal is made piecewise constant by dividing the control signal into smaller control intervals. For direct collocation, the control interval can then be divided into collocation intervals that is further divided by pre-defined collocation points [33]. An illustrative figure of how the intervals are divided can be seen in Figure 2.2 [29]. During each collocation interval the state-trajectory is approximated using an interpolating polynomial written on Lagrange-form, whose differentiated value is forced to match the OCP state derivative in equation 2.1 at each collocation point. At the end of the collocation interval the state value and the derivative is set to the value of the polynomial at that point as a continuity constraint. Then the interpolating polynomial is integrated with respect to the independent variable t to give an approximation of the Lagrange term in the OCP [29]. According to Lars Eriksson at Linköping University (2022), direct collocation efficiently handles constraints but requires the problem to be continuously differentiable due to its gradient based methodology.

The polynomial can then be integrated with respect to the independent variable, the time t for this project. The method is described below.

The time interval $t \in [0, T]$ gets discretised into K equidistant segments with the length h representing the control interval where the control signal is approximated as piecewise constant:

$$h = \frac{T}{K} \tag{2.3}$$

where

$$\begin{aligned} &t_{1,0,0} = 0 \\ &t_{k+1,0,0} = t_{k,0,0} + h, \text{ for } k = 1, ..., K. \end{aligned} \tag{2.4}$$

$$u(t) = u_k \text{ for } t \in [t_{k,0,0}, t_{k+1,0,0}] \tag{2.5}$$

The control segment is divided into N collocation interval with the length h_k:

$$h_k = \frac{h}{N} \tag{2.6}$$

where

$$\begin{aligned} t_{k,i,0} = t_{k,0,0} + ih_k \text{ for } & k = 1, ..., K \\ & i = 0, ..., N \end{aligned} \tag{2.7}$$

Each collocation interval $d + 1$ is equipped with the predefined collocation points τ:

$$\begin{aligned} t_{k,i,j} = t_{k,i,0} + \tau_j h_k \text{ for } & k = 1, ..., K \\ & i = 0, ..., N - 1 \\ & j = 0, ..., d \end{aligned} \tag{2.8}$$

where $\tau_0 = 0$.

On each collocation interval the state trajectory is approximated using the interpolation polynomial:

$$\begin{aligned} &t = t_{k,i,0} + h_k \tau, \text{ for } t \in [t_{k,i,0}, t_{k,i+1,0}] \\ &x(t) \approx \sum_{j=0}^{d} L_j(\tau) x_{k,i,j} \end{aligned} \tag{2.9}$$

where

$$L_j(\tau) = \prod_{r=0, r\neq j}^{d} \frac{\tau - \tau_r}{\tau_j - \tau_r} \tag{2.10}$$

and $x_{k,i,j}$ are the polynomial coefficients. To force the polynomial to equality at the collocation points the trajectory is differentiated as:

$$f(t_{k,i,j}, x_{k,i,j}, u_k, P) = \frac{1}{h_k} \sum_{r=0}^{d} \frac{\partial L_r}{\partial \tau}(\tau_j) x_{k,i,r} \text{ for } \begin{matrix} k = 1, .., K \\ i = 0, ..., N-1 \\ j = 1, ..., d \end{matrix} \tag{2.11}$$

The state value is obtained by evaluating the polynomial at the end of the interval:

$$x^+_{k,i} = \sum_{r=0}^{d} L_r(1) x_{k,i,r} \text{ for } \begin{matrix} k = 1, ..., K \\ i = 0, .., N-1 \end{matrix} \tag{2.12}$$

where

$$x^+_{k,i} = \begin{cases} x_{k+1,0} \text{ if } i = N-1 \\ x_{k+1,0} \text{ otherwise} \end{cases} \tag{2.13}$$

Then the Lagrange term is integrated by using quadrature:

$$\int_0^T L(t, x(t), u(t), P) dt \approx \sum_{k=1}^{K} \sum_{i=0}^{N-1} h_k \sum_{r=0}^{d} \int_0^1 L_r(\tau) d\tau L(t_{k,i,r}, x_{k,i,r}, u_k, P) \tag{2.14}$$

For an illustration of the state trajectory interpolation, and of how the final NLP can be expressed interested readers are referred to [29, 33].

2.3.4 PID

PID stands for Proportional, Integral, Derivative and is a very intuitive controller to understand and tune by hand. Comparing results of a different controller to a PID-controller can give a very good view of how good the more complex controller is compared to the time it takes to develop. The control signal can be expressed as

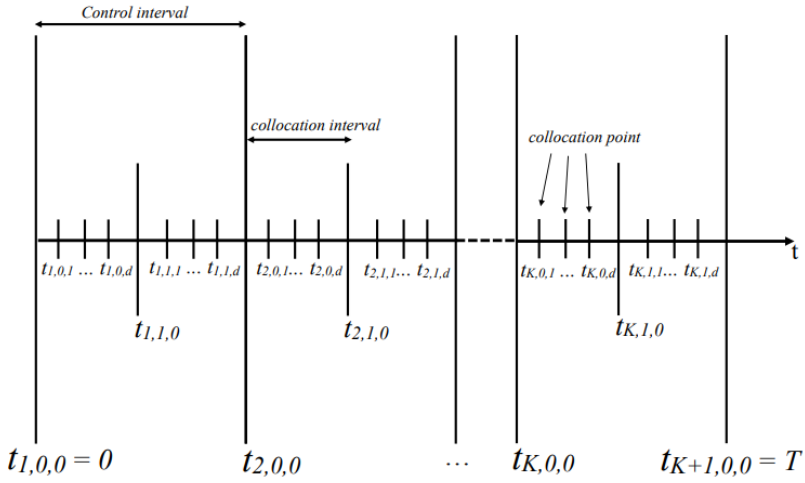


Figure 2.2: *Illustration of the divided control interval.*

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{2.15}$$

where t is the time, τ is the time in the integral, K_p, K_i and K_d are gain coefficients for the respective term. $e(t)$ is the error from the output signal $y(t)$ compared to the reference signal $r(t)$, i.e. the feedback signal.

$$e(t) = r(t) - y(t) \tag{2.16}$$

The PID-controller does not require the use of the feedback signal, it can instead use solely the reference signal when calculating the control signal, it is however harder then to obtain a small error between the reference and output signal.

The different gains can be zero, in practice turning the PID to a P, PI or PD-controller depending on what the specific problem requires the controller to accomplish.

2.3.5 YOP

YOP, also know as "yet another optimal control problem", is an optimisation tool based on CasADi [34] in MATLAB [35] and developed by Viktor Leek [29]. YOP

utilises collocation methods to find the optimal values for the control signal based on the given objective function. When using YOP, the problem transcripts from an OCP to NLP. Then an interior point method (IPOPT) in CasADi solves the NLP. To use YOP on an engine-model, the system needs to be modeled in a MATLAB function-script that will be regulated by a control script. In the control script the regulator is declared, such as initial and final values for the states, control signals, constraints and a simulation that run the optimisation resulting in an optimal path for the control signal given the objective function. Different initial values can be put into the script in order to replicate the operational environment for different transients of the system. When running the optimisation it will either converge to a solution or not. If the problem does not converge, the problem might be over-constrained, bad initial guesses or that the model is ill defined resulting in extreme values. When the problem is built and the solution converged, the command window in MATLAB will display the IPOPT output containing number of iterations, unscaled objective value at current point, the barrier parameter μ among others together with other interesting parameters such as CPU calculation time. The collocation points are by default set to Radau points.

3

Data Collection and Experiments

The data used in this project were attained from AUROBAY mainly consisting of two different data sets called "Time Independent" and "Timeseries". The data tests were performed at a test track in Gothenburg on a VEP4 gen3 LP-miller engine for the Time independent and timeseries data. The cylinder data were performed on a MP-miller engine.

3.1 Time Independent

The Time Independent data set are a smaller set of data collected at different steady state points for four different ambient conditions. This data is used to create the static models. Data for $p2/p1 < 1.1$ is discarded since the turbo is barely doing any work in this operating region and the models are unreliable in this region. The time step for the data were chosen by convenience to 0.01 seconds with the chosen SIMULINK solver is a *discrete* solver of *fixed-step* type. This setting is the same for each data set running at a time step of 0.01 seconds

3.2 Timeseries

The timeseries data are a larger set of data collected during a driver defined drive cycle performed by the project's supervisor Bohan from AUROBAY. The data is time dependent and each value corresponds to a certain time of the cycle, and thus suitable to use when simulating a complete system and creating dynamic models. See Figure 3.1 for an illustration of the cycle, and Figure 3.2a, 3.2b and 3.2c for each transient. The marked areas of the figure are the transients that have been mimicked in YOP above in section 5.2.1. When simulating, the data

for the first 34000 values are discarded due to the engine not operating properly until then. The time step of the data is 0.01 sec.

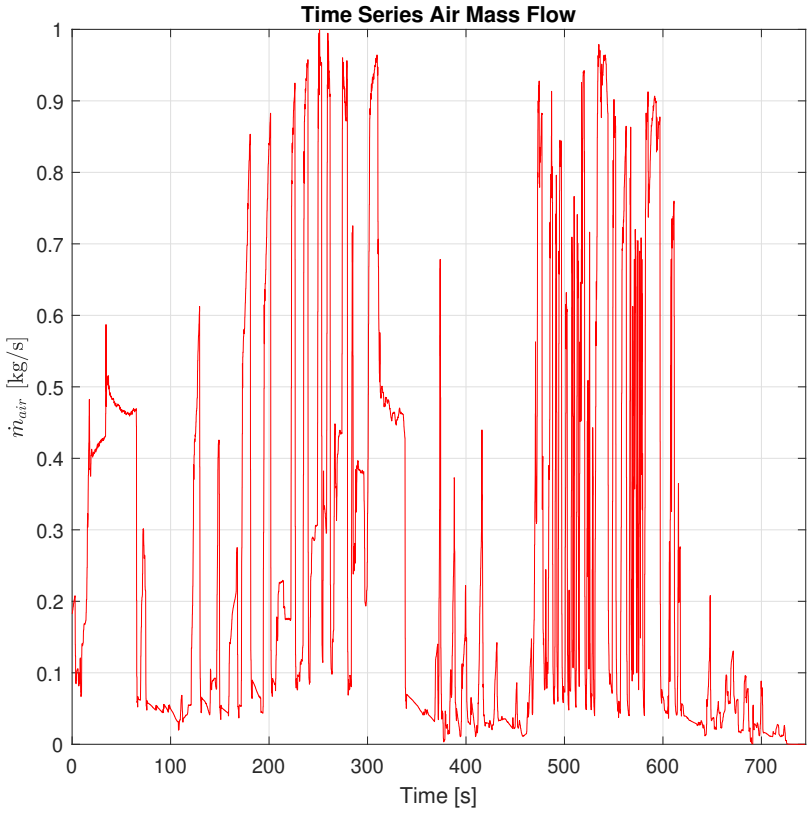
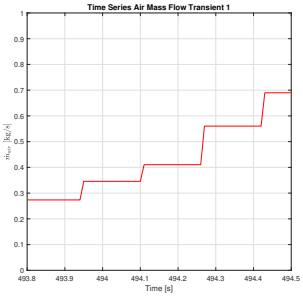


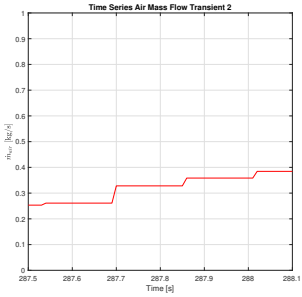
Figure 3.1: *Plot of the $\dot{m}_{air}$ for the complete timeseries cycle.*

3.3 Cylinder Data

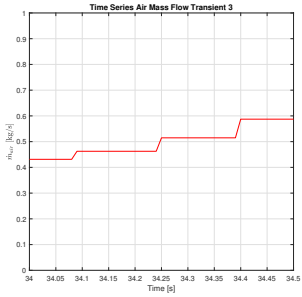
The cylinder data set used is containing all parameters needed to build a cylinder model. What is special from the other sets is that this contains data for the volumetric efficiency, BSFC and engine torque. The data is time independent and received from the project's supervisor Bohan from AUROBAY. The time step for the data were chosen by convenience to 0.01 sec.



(a) Plot of the $\dot{m}_{air}$ for transient 1.



(b) Plot of the $\dot{m}_{air}$ for transient 2.



(c) Plot of the $\dot{m}_{air}$ for transient 3.

Figure 3.2: *Illustrations of the three different transients that have been mimicked and used in YOP as a reference, extracted from Figure 3.1 for readability.*

4

Modelling of VNT and Cylinders

As previously stated, the models are map-based MVEM models with neglected control volumes. The model built in this thesis consists of one state, turbo speed N_{turbo}, and one control signal, VNT position x_{vnt}. The compressor and turbine are modelled as static with a set of model-equations primarily from [27], as for the dynamic models. The cylinder model is modelled as static, different equations for the volumetric efficiency in the cylinder were tested before one proved to give superior results in validation. Due to the neglecting of the control volumes there is no rate of change between the compressor and cylinder, as well as between the cylinder and turbine. The maps mentioned have been created by using the MATLAB functions "`meshgrid()`", with vectors of the map's respective x- and y-value as inputs, and "`griddata()`" with the vectors of x-, y-, z-values as well as the outputs from the previous function as input [35]. The SIMULINK scheme over the equations can be seen in Appendix A.

4.1 Compressor Models

The massflow through the compressor and the speed of the compressor are considered as normalised corrected quantities and are thus able to handle changes in inlet conditions.

$$\dot{m}_{c,corr} = \frac{\dot{m}_c \sqrt{\frac{T_1}{T_{ref}}}}{\frac{p_1}{p_{ref}}} \tag{4.1}$$

$$N_{c,corr} = \frac{N_c}{\sqrt{\frac{T_1}{T_{ref}}}} \tag{4.2}$$

These two values can then be used to create maps for the compressor efficiency and the pressure ratio over the compressor. With measured data for the intake pressure and boost pressure the pressure ratio can be calculated. The pressure ratio map can then be built with the output from equation (4.3) and the corrected values.

$$\Pi_c = \frac{p_2}{p_1} \tag{4.3}$$

If the pressure ratio is known, then the boost pressure p_2 can be calculated.

$$p_2 = \Pi_c * p_1 \tag{4.4}$$

The efficiency is more complex to calculate. One way of doing it is to calculate the observed efficiency, i.e. the ratio between the ideal and actual power output of the compressor. With the output from equation (4.5) and the corrected values the compressor efficiency map can be built. With the output from 4.5, the pressure-ratio and measured inlet temperature the temperature of the boosted air (T2) can be modelled in a simulation.

$$\eta_c = \eta_{c,obs} = \frac{\dot{W}_{c,ideal}}{\dot{W}_c} = \frac{\cancel{\dot{m}_{air} c_{p,air}} T_1 \left(\left(\frac{p_2}{p_1} \right)^{\frac{\gamma_{air}-1}{\gamma_{air}}} - 1 \right)}{\cancel{\dot{m}_{air} c_{p,air}} (T_2 - T_1)} \tag{4.5}$$

$$T_2 = T_1 \left(\frac{\Pi_c^{\frac{\gamma_{air}-1}{\gamma_{air}}} - 1}{\eta_c} + 1 \right) \tag{4.6}$$

Now the power output from the compressor $\dot{W}_c$ can be modelled.

$$\dot{W}_c = \dot{m}_{air} c_{p,air} (T_2 - T_1) \tag{4.7}$$

From these compressor equations, the following maps can be built and used in calculations with measured data. The maps have been created by calculating the values from the equations with the time independent data set. This data as well as the set of timeseries data from AUROBAY have then been used as input to the maps and models in SIMULINK to validate the output of each part of the system. Since neither of these have been calculated with the input data, they are be considered grey boxes. In SIMULINK, the maps have been given the settings

to "clip" the output if any of the inputs goes beyond what is defined for the map. The SIMULINK schemes of the compressor can be seen in Appendix A.1.

$$\Pi_{c,map} = \Pi_c(\dot{m}_{c,corr}, N_{c,corr}) \tag{4.8}$$

$$\eta_{c,map} = \eta_c(\dot{m}_{c,corr}, N_{c,corr}) \tag{4.9}$$

4.2 Turbine Models

To find the effective area of the gasoline particle filter (GPF), the general equation for the mass flow on incompressible turbulent restriction on a pipe is applied:

$$\dot{m} = A_{effective}\sqrt{\frac{p_{us}}{R_{exh}T_{us}}}\sqrt{\Delta p} = A_{effective}\sqrt{\frac{p_{us}}{R_{exh}T_{us}}}\sqrt{p_{us} - p_{ds}} \tag{4.10}$$

where the indices *us* represent upstream and *ds* represent downstream and R is the ideal gas constant. If everything but the value of the effective area are known, equation (4.10) can successfully be used to calculate the missing value. If the rear effective area of the GPF is to be calculated, the equation are rewritten with the ambient pressure p_{amb} as the downstream pressure.

$$A_{eff,rear} = \sqrt{\frac{4\dot{m}_{exh}^2 R_{exh} T_{GPF,ds}}{(2p_{GPF,ds} - p_{amb})^2 - p_{amb}^2}} \tag{4.11}$$

The effective area of the front of the GPF can be calculated in the same way but with the pressures and temperatures shifted one step further upstream towards the engine.

$$A_{eff,front} = \sqrt{\frac{4\dot{m}_{exh}^2 R_{exh} T_4}{(2p_4 - p_{GPF,us})^2 - p_{GPF,us}^2}} \tag{4.12}$$

Equation (4.12) can also be rewritten as a model for p_4 if the effective area is known.

The effective area of the turbine is the main parameter that is affected by the control signal for the VNT vanes in the model. From equation (4.17) one can see that a change in the effective area will also lead to a change in $\dot{m}_{exh}$ and the pressures around the turbine. Ψ is the choke function and has the pressure ratio p_r as input as is calculated with the following equations. The square root in equation 4.13 derives from the fact that two equal stages of expansion are assumed in the

turbine, one in the stator and one in the rotor. The two equal expansions with 50% degree of reaction generate the square root.

$$p_r = \sqrt{\frac{p_4}{p_3}} \tag{4.13}$$

$$\Pi(p_r) = MAX\left(pr, \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}}\right) \tag{4.14}$$

$$\Psi_0(\Pi) = \sqrt{\frac{2\gamma}{\gamma-1}\left(\Pi^{\frac{2}{\gamma}} - \Pi^{\frac{\gamma+1}{\gamma}}\right)} \tag{4.15}$$

$$\Psi(p_r) = \Psi_0(\Pi(p_r)) \tag{4.16}$$

$$A_{eff} = \dot{m}_{exh}\frac{\sqrt{RT_3}}{p_3 * \Psi(p_r)} \tag{4.17}$$

By combining the position of the VNT vanes, X_{vnt}, with the turbine speed parameter (TSP), a map of the effective area can be created with very good accuracy. TSP can be calculated as follows.

$$TSP = \frac{N_t}{\sqrt{T_3}} \tag{4.18}$$

The upstream pressure p_3 from the turbine can be calculated by:

$$p_3 = \frac{Z}{\Psi} \tag{4.19}$$

where Z is a parameter from the project's supervisor Bohan Liang (2022), made up by rearranging the modified compressible flow equation

$$\dot{m}_{exh} = \frac{p_3}{\sqrt{RT_3} \cdot A_{eff}(X_{vnt}, TSP) \cdot \Psi(p_r)} \tag{4.20}$$

to

$$Z_{\Psi,p_3} = \frac{\dot{m}_{exh}\sqrt{RT_3}}{A_{eff}(X_{vnt}, TSP)} \rightarrow Z_{\Psi,p_3} = p_3\Psi(p_r) \tag{4.21}$$

A map for the pressure p_3 can be created with good accuracy by combining Z and p_4.

The ideal turbine power can be calculated by with the following equation:

$$\dot{W}_{t,ideal} = \dot{m}_{exh} c_{p,exh} T_3 \left(1 - \left(\frac{p4}{p3} \right)^{\frac{\gamma_{exh}-1}{\gamma_{exh}}} \right) \tag{4.22}$$

The efficiency of the turbine can be calculated by dividing the ideal power with the power of the compressor as follows.

$$\eta_t = \frac{\frac{\dot{W}_c}{\eta_m}}{\dot{W}_{t,ideal}} \tag{4.23}$$

η_m is the mechanical efficiency of the turbo. It is generally a difficult parameter to estimate, however, it is possible to omitt η_m by assuming it is included in η_t.

The Blade Speed Ratio (BSR) is a common parameter to use in a map for the turbine efficiency η_t. If the engine speed also is used in the map, it is possible to handle exhaust pulse compensation.

$$BSR = \frac{N_t r_t}{\sqrt{2 c_{p,exh} T_3 \left(1 - \left(\frac{p4}{p3} \right)^{\frac{\gamma_{exh}-1}{\gamma_{exh}}} \right)}} \tag{4.24}$$

where r_t is the radius of the turbine.

With η_m omitted as a part of η_t, the actual power output from the turbine can be calculated.

$$\dot{W}_t = \eta_t \dot{W}_{t,ideal} \tag{4.25}$$

From these turbine equations, the following tables and maps can be built and used in calculations and simulations with measured data. The maps have been created by calculating the values from the equations with the time independent data set. This data as well as the set of timeseries data from AUROBAY have then been used as input to the maps and models in SIMULINK to validate the output of each part of the system. $A_{eff,map}$ and $\eta_{t,map}$ are be considered grey boxes. In SIMULINK, the maps have been given the settings to "clip" the output if any of the inputs goes beyond what is defined for the map. The SIMULINK schemes of the turbine model can be seen in Appendix A.2.

$$A_{eff,rear,table} = A_{eff,rear}(\dot{m}_{exh}) \tag{4.26}$$

$$A_{eff,front,table} = A_{eff,front}(\dot{m}_{exh}) \tag{4.27}$$

$$A_{eff,map} = A_{eff}(X_{vnt}, TSP) \tag{4.28}$$

$$p_{3,map} = p_3(Z, p_4) \tag{4.29}$$

$$\eta_{t,map} = \eta_t(BSR, N_{eng}) \tag{4.30}$$

4.3 Cylinder Models

The volumetric efficiency of the cylinders in the engine can be calculated with the following equation from [27].

$$\eta_{vol} = \frac{\dot{m}_{air} n_r}{\rho_{air} V_D N_{eng}} \tag{4.31}$$

The ideal gas law is used to calculate the density of the air, with the amount of moles expressed as the mass divided by the molar mass.

$$\rho_{air} = \frac{m}{V_D} = \frac{M p_2}{R_{air,mol} T_2} \tag{4.32}$$

As previously stated in chapter 2.1.3, it is extra interesting for the project to know how η_{vol} is affected by the pressure ratio between p_2 and p_3. This pressure ratio together with the engine speed can create a black box map for the volumetric efficiency.

From [27], the air mass flow in to the cylinders can be calculated.

$$\dot{m}_{air} = \frac{V_D p_2 \frac{N_e}{60} \eta_{vol}}{T_2 n_r R_{air}} \tag{4.33}$$

The mass flow $\dot{m}_{air}$ is expressed in kilograms per second in equation (4.33). Another typical unit for the mass flow that also relates more clearly to the load on

the engine is milligrams per stroke. The equation below shows how to convert the mass flow to this unit for a four-stroke engine.

$$m_{stroke} = \left(\frac{\dot{m}_{air} * 1e6}{\frac{N_{eng}}{60}} \right) * \frac{1}{2} \tag{4.34}$$

The exhaust temperature T_3 is very hard to measure in practice. However, it can be approximated as a function of the air mass flow in to the cylinder. Because of the characteristics of the temperature, it is modelled as a fifth-grade polynom below a certain value for the mass flow and as a linear equation above this value.

$$T_3 = \begin{cases} x_1 + x_2\dot{m}_{exh} + x_3\dot{m}_{exh}^2 + x_4\dot{m}_{exh}^3 + x_5\dot{m}_{exh}^4 + x_6\dot{m}_{exh}^5, & \text{if } \dot{m}_{exh} < 0.14 \text{ kg/s} \\ x_7 + x_8\dot{m}_{exh}, & \text{otherwise} \end{cases} \tag{4.35}$$

For the same argument as with the volumetric efficiency, the brake specific fuel consumption (BSFC) can also be mapped with a black box approach of the pressure ratio and engine speed. To create the map, a function of BSFC can be fitted from measured data of the pressure ratio, engine speed and calculated BSFC.

The following maps can be created from the cylinder equations and used in simulations of the turbocharger. The maps and apporixmated T_3-function have been created from the cylinder data set from AUROBAY. The maps and the function have then been validated with this data as well as the timeseries data to validate them in a SIMULINK simulation. None of the outputs from the cylinder maps have been calculated with their respective inputs, but there is still a proper theoretical background to how they should look depending on the map inputs and they are therefore considered to be grey boxes. In SIMULINK, the maps have been given the settings to "clip" the output if any of the inputs goes beyond what is defined for the map. The SIMULINK schemes of the cylinder model can be seen in Appendix A.3.

$$\eta_{vol,map} = \eta_{vol}(\frac{p_2}{p_3}, N_e) \tag{4.36}$$

$$BSFC_{map} = BSFC(\frac{p_2}{p_3}, N_e) \tag{4.37}$$

4.4 Dynamic Models

To simulate the entire system, the speed of the turbocharger is needed as a feedback into the models. The turbo speed is calculated using the previously calculated value together with the integrated derivative of the speed. The derivative

depends on the imbalance of the generated power from the turbine and the consumed power of the compressor. To calculate the derivative of the speed, the following equation is used.

$$\dot{N}_{turbo,k} = K_{inertia}\left(\frac{\dot{W}_{t,k-1} - \dot{W}_{c,k-1}}{N_{turbo,k-1}} - M_{fric}(N_{turbo,k-1})\right) \tag{4.38}$$

where $K_{inertia}$ is a constant gain used as a tuning parameter and is a representation of the turbo inertia and the converting of speed from rpm to rad/s. M_{fric} is a merge of different uncertainties that occur primarily at high turbo speeds. The uncertainties are not included in any of the models and M_{fric} is a table fitted by hand. The table is build with engine speed as x-values and the steady state error between the compressor and turbine as y-values to stabilise the derivative.

The derivative is then passed through a low-pass filter (LP-filter) and multiplied by the time step (Ts) to gain a value that should be added or removed from the speed of the last iteration. The LP-filter was created in continuous time and transferred to discrete time with help of the MATLAB-function "`c2d()`".

$$LP = c2d\left(\frac{4}{0.2s + 4}\right) = \frac{0.1813}{z - 0.8187} \tag{4.39}$$

$$N_{turbo,k+1} = N_{turbo,k} + LP(\dot{N}_{turbo,k+1})T_S \tag{4.40}$$

To validate the dynamic models, the timeseries data set have been used as input in SIMULINK to simulate the entire turbocharger model. The turbo speed N_{turbo} from the simulation have then been compared to the speed from the data. The SIMULINK scheme of the dynamic model can be seen in Appendix A.4.

5

Control Strategy for the VNT Vanes

For the control of the VNT, two different controllers have been developed. The primary control strategy is using collocation and utilises the previously mentioned YOP to find the optimal path for the control signal. The second control strategy is a more simple PID, this controller was developed primarily as a comparison for the quality of the collocation strategy. The primary objective of the control strategy is to properly control the positive tip-in transients. Tip-in is when the driver is starting to press the accelerator and the car is accelerating while tip-out is when the driver lets go of the accelerator and the speed of the car decreases.

5.1 Using a PID-controller

A PID-controller can be implemented onto the system with ease. This is done by having the entire turbo-model in SIMULINK as a subsystem. From the subsystem, the cylinder air flow is the only output and this is compared to a reference value of the air flow. This error signal is then the input signal to the PID-controller which in turn outputs a control signal for the position of the VNT vanes. The reference signal consists of a hand-made transient. This transient can have a different appearance based on what should be tested, e.g. a normal increase in air flow, steady state of the air flow, or a step in the higher regions of the air flow. To tune the controller, the K_p, K_i and K_d gain values are needed to be tuned by hand or by a tuning tool in MATLAB. Tuning by hand is done by looking at a normal transient of the system and depending on the characteristics of the resulting output signal, increase or decrease the gain that primarily affect the characteristics that need to be altered. In the built-in tuner tool from MATLAB, the user can choose how fast the response time should be, i.e. how much overshoot or

undershoot the system should have and how fast the system should be compared to how robust it should be. From these specific characteristics, the built-in tuner then gives optimised values for the gains of the PID-controller. A good reference following during the tip-in is desired. The SIMULINK model of the PID-controller can be seen in Figure 5.1.

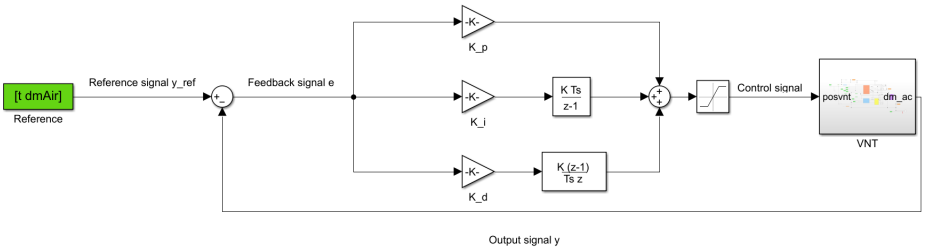


Figure 5.1: *SIMULINK scheme of the PID-controller. The turbo model is inside the "VNT"-subsystem block.*

5.2 Using Collocation Methods

The direct collocation control strategy is built using a toolbox created for MATLAB, called "YOP." All of the equations used to build the SIMULINK model have been implemented, adapted and defined as a function in one of three scripts dedicated for YOP in MATLAB, called "turbo-model". The adaption is due to the fact that YOP can not read the created maps for SIMULINK and instead had to be fitted and defined as functions of varying types using the "Curve Fitting Toolbox" in MATLAB, see Appendix C for illustrations of the adaptations. Inputs to the function are the state and control signal, the engine constants and initial values for parameters are called in from the second of the three scripts, called "turbo-parameters" which is further explained below. The state is turbo speed N_{turbo} and control signal X_{vnt}. The function finally calculates the ODE, same as equation (4.38) and relevant parameters are defined as outputs from the function, making them available in the third script, "turbo-control".

In the "turbo-parameter" script, all of the engine constants and initial parameter values extracted from different transients from the timeseries data are defined to be read by the "turbo-model" script.

The third script "turbo-control" is the script that build and define how YOP ought to control the optimisation and runs the simulation. It is defined in three sections. The first section defines the state and control signal and their corresponding max and min values before it calls the "turbo-model" script that runs the values from the "turbo-parameter" script mentioned earlier. The independent variable time (t) is created and given the variables $t0$ and tf representing initial and final time of the simulation. In the second section an initial guess is created. This allows the

user to see if the system is unstable and the magnitude of the state and control signal. The solver is defined as "idas" (DAE solver in MATLAB). In the third section the OCP is defined with the objective function and corresponding constraints for the problem of interest and solved using the earlier created initial guess and the solver "ival" with 100 control intervals.

5.2.1 The Studied OCP

For the project, positive transients are the most relevant transients to examine, but alternative transients are also of interest. The air mass flow will be used as a reference in some of the studied transients since it correlates with the demanded load from the driver. The OCPs that will be studied are:

- minimise the error between a reference representing three different transients. The first one representing a transient from approximately 300 mg/stroke to 900 mg/stroke (low to high air mass flow). The second one, a transient from approximately 300 mg/stroke to 600 mg/stroke (low to medium air mass flow) and the third one, approximately a 600 mg/stroke to 900 mg/stroke transient (medium to high air mass flow), see section 3.

  - Transient 1, 300 mg/stroke to 900 mg/stroke:

$$
\begin{aligned}
\min_{\dot{m}_{air,error}} \quad & 100 \int_{t_0}^{t_f} (\dot{m}_{air} - (0.044 - 0.0394t + 0.38t^2 - 0.288t^3))^2 \\
\text{s.t.} \quad & x_{min} \leq x \leq x_{max}, \\
& u_{min} \leq u \leq u_{max} \\
& p_{3,min} \leq p_3 \leq p_{3,max} \\
& \eta_{t,min} \leq \eta_t \leq \eta_{t,max} \\
& t_0 = 0.1 \\
& t_f = 0.8 \\
& x(t_0) = x0 \\
& \dot{x} = dx
\end{aligned}
\tag{5.1}
$$

– Transient 2, 300 mg/stroke to 600 mg/stroke:

$$\begin{aligned} \min_{\dot{m}_{air,error}} \quad & 100 \int_{t_0}^{t_f} (\dot{m}_{air} - (0.03884 + 0.02192t + 0.1147t^2 - 0.1529t^3))^2 \\ \text{s.t.} \quad & x_{min} \leq x \leq x_{max}, \\ & u_{min} \leq u \leq u_{max} \\ & p_{3,min} \leq p_3 \leq p_{3,max} \\ & T_2(t_0) \geq 300 \\ & t_0 = 0 \\ & t_f = 0.6 \\ & x(t_0) = x0 \\ & \dot{x} = dx \end{aligned} \tag{5.2}$$

– Transient 3, 600 mg/stroke to 900 mg/stroke:

$$\begin{aligned} \min_{\dot{m}_{air,error}} \quad & 100 \int_{t_0}^{t_f} (\dot{m}_{air} - (0.014sin(4t + 2.387) + 0.079))^2 \\ \text{s.t.} \quad & x_{min} \leq x \leq x_{max}, \\ & u_{min} \leq u \leq u_{max} \\ & p_{3,min} \leq p_3 \leq p_{3,max} \\ & t_0 = 0.6 \\ & t_f = 1.2 \\ & x(t_0) = x0 \\ & \dot{x} = dx \end{aligned} \tag{5.3}$$

- minimise the error between a reference with both positive and negative transient, "tip-in - tip-out". The same OCP as transient 3 will be used, but for a longer t_f.

$$
\begin{aligned}
\min_{\dot{m}_{air,error}} \quad & 100 \int_{t_0}^{t_f} (\dot{m}_{air} - (0.014 sin(4t + 2.387) + 0.079))^2 \\
\text{s.t.} \quad & x_{min} \leq x \leq x_{max}, \\
& u_{min} \leq u \leq u_{max} \\
& p_{3,min} \leq p_3 \leq p_{3,max} \\
& t_0 = 0.6 \\
& t_f = 2 \\
& x(t_0) = x0 \\
& \dot{x} = dx
\end{aligned}
\tag{5.4}
$$

- maximise the volumetric efficiency while reaching a mass flow of 900 mg/stroke, using the same initial guess as for transient 3.

$$
\begin{aligned}
\max_{\eta_{vol}} \quad & \int_{t_0}^{t_f} \eta_{vol}^2 \\
\text{s.t.} \quad & x_{min} \leq x \leq x_{max}, \\
& u_{min} \leq u \leq u_{max} \\
& p_{3,min} \leq p_3 \leq p_{3,max} \\
& \eta_{t_min} \leq \eta_t \leq \eta_{t,max} \\
& \dot{m}_{air}(t_f) \geq 900 \\
& t_0 = 0 \\
& t_f = 0.6 \\
& x(t_0) = x0 \\
& \dot{x} = dx
\end{aligned}
\tag{5.5}
$$

- minimise the time taken to reach a mass flow of 900 mg/stroke. The initial guess is the same one as for transient 3. The OCP will be solved with both a free t_f to illustrate the optimal control strategy, and a $t_f \geq 0.6$ sec. for comparison with the results from transient 3.

$$
\begin{aligned}
\min_{t} \quad & t_f \\
\text{s.t.} \quad & x_{min} \leq x \leq x_{max}, \\
& u_{min} \leq u \leq u_{max} \\
& p_{3,min} \leq p_3 \leq p_{3,max} \\
& \dot{m}_{air}(t_f) \geq 900 \\
& t_0 = 0 \\
& t_f \geq 0 \\
& x(t_0) = x0 \\
& \dot{x} = dx
\end{aligned}
\tag{5.6}
$$

The described model script, parameter script and the control script used to run YOP in MATLAB can be found in Appendix B.

6
Results

In this part, the results from analysing the questions from chapter 1.3 will be presented.

6.1 Modelling of the Turbocharger

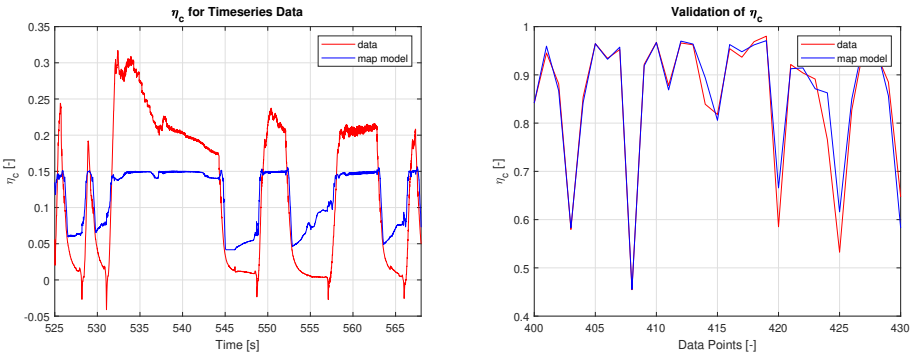
Note that the compressor and cylinder model were primarily developed in SIMULINK by a closely related and simultaneous project, [36]. That project is also in cooperation with AUROBAY with the purpose of looking at the possibility of using an electric turbocharger where energy from the turbine can be stored in a battery. All of the presented results using timeseries data are simulated with the complete engine model connected, see Appendix A.14. The results using the time independent data are simulated with each component isolated. The results presented with the timeseries data display only a sequence of the drive cycle presented in chapter 3 for readability. To highlight model error, the parameters with poor timeseries data performance are presented along the results from the data the model was built upon.

Due to a confidential agreement, all of the values presented in the figures have been normalized.

6.1.1 Compressor

The compressor model consists of the equations presented in chapter 4.1. In Figure 6.1a, the model for the compressor efficiency, η_c, can be seen. The model follows the calculated value very well in transients but appears to be getting capped where the calculated value from the data keeps increasing or decreasing. In Fig-

ure 6.1b, it can be seen that the model follows the efficiency calculated from the given data it was built upon with very few errors.



(a) *η_c for the SIMULINK model and calculated value from the timeseries data.*

(b) *η_c from the SIMULINK model and calculated value from the time independent data the model was built upon.*

Figure 6.1: *Validation of η_c for the two data sets.*

The compressor pressure ratio Π_c can be seen in Figure 6.2. The simulated value from the model follows the data very well across the entire spectrum of values. However, the model often give a too high value with a steady state error for high pressure ratios.

In Figure 6.3 and 6.4, the resulting maps from equation (4.9) and (4.8) respectively can be seen.

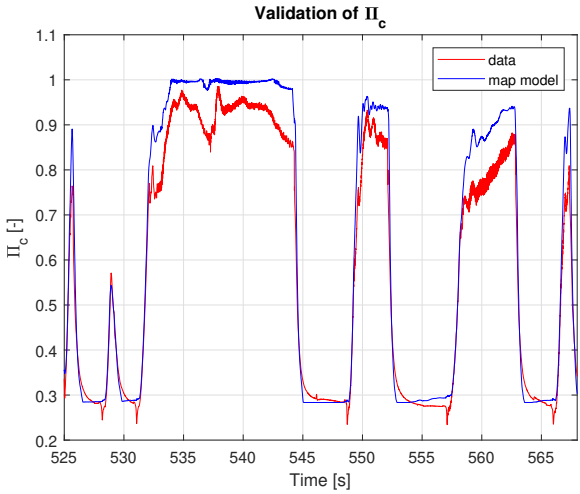


Figure 6.2: *Plot of Π_c given from the timeseries data and simulated from the model.*

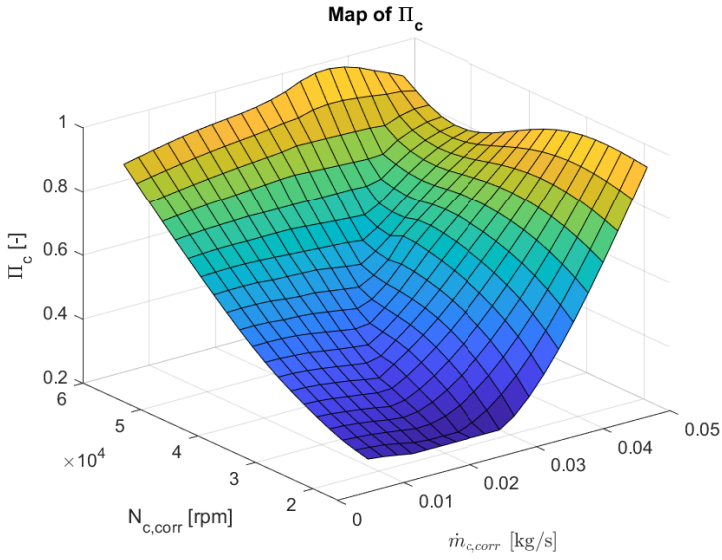


Figure 6.4: *Plot of the map for Π_c.*

6.1.2 Turbine

The turbine model consists of the equations presented in chapter 4.2. The effective rear and front area of the GPF respectively can be seen in Figure 6.5 and 6.6. There is no data for these two parameters in the timeseries data set and therefore

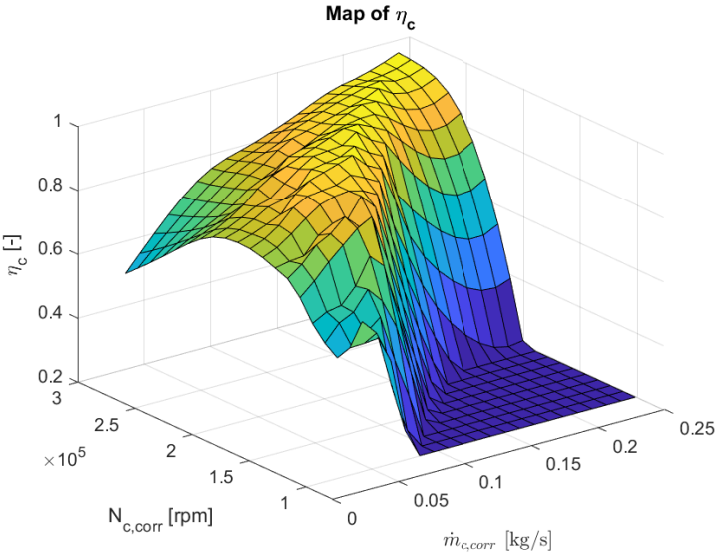


Figure 6.3: *Plot of the map for η_c.*

the inputs are the same as the values the tables have been built on.

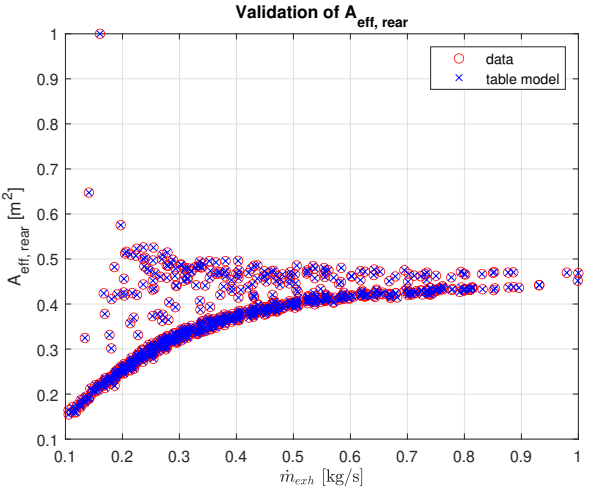


Figure 6.5: *Plot of $A_{eff,rear}$ given from the time independent data and simulated from the model.*

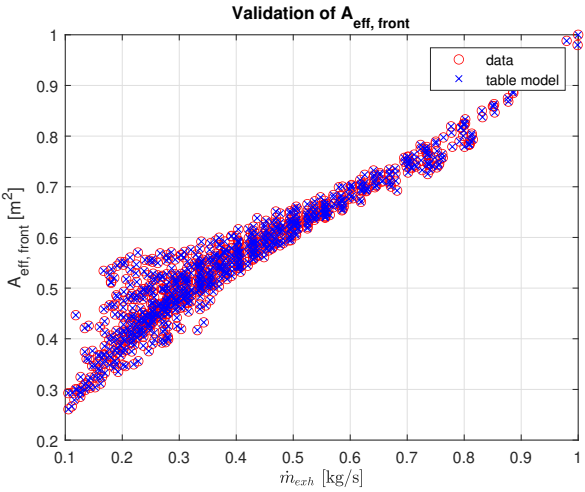


Figure 6.6: *Plot of $A_{eff,front}$ calculated from the time independent data and simulated from the model.*

As with the effective areas of the GPF, the effective area of the turbine, A_{eff}, have also been simulated with the same data the map was built upon. However, the inputs to the map are not the same parameters as what was used to calculate the area, as can be seen in equation (4.17) and Figure 6.10. The model in Figure 6.7 works very well, especially for the lower values of the area. The more the area increases, the more insecure the model becomes and at the greatest values of the area, the model gives a couple rather odd values.

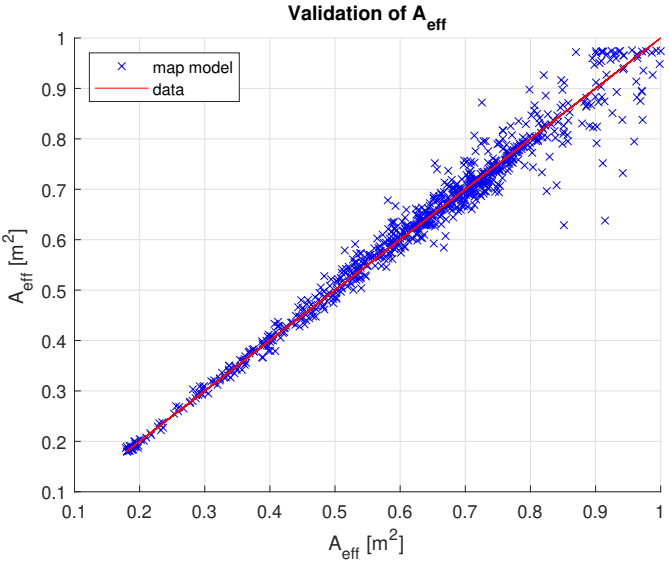


Figure 6.7: *Plot of A_{eff} calculated from the time independent data and simulated values from the model.*

The model of p_3 works very well across all values of p_3 from the timeseries data, as can be seen in Figure 6.8. The given data is very oscillative due to a missing filter when the measurements were taken but the model still seems follow the general dynamics of the data very well.

In Figure 6.9, the turbine efficiency η_t calculated with the time independent data can be seen together with the results of two different maps for the efficiency. The blue model is from AUROBAY and capped at a reasonable value. The green model was developed during the project but is not capped. In order to obtain more realistic and better outputs from the model, the blue model have been used during the simulations. None of the maps however appear to properly catch the dynamics of the data.

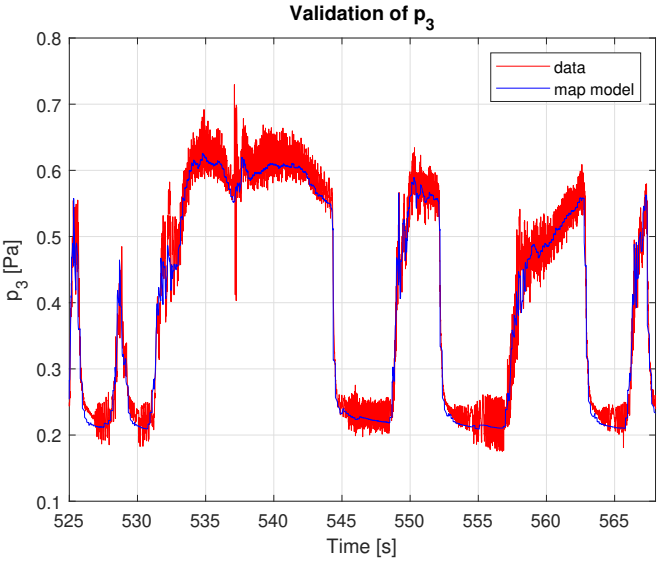


Figure 6.8: *Plot of p_3 given from the timeseries data and simulated from the model.*

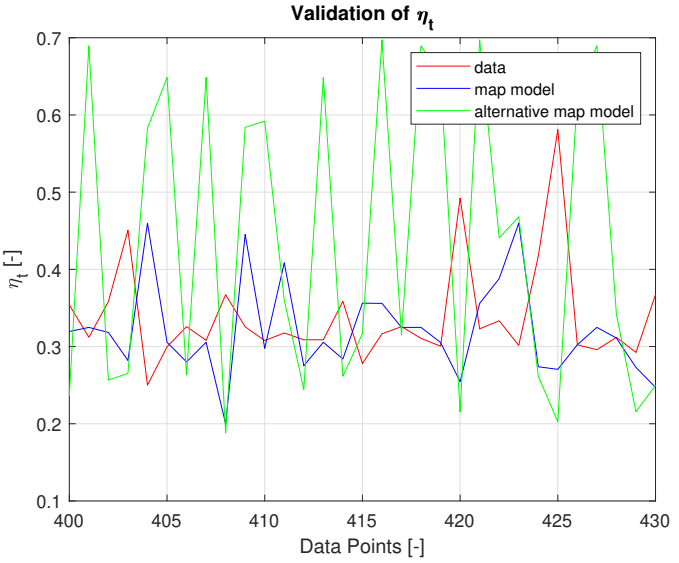


Figure 6.9: *Plot of η_t calculated from the time independent data and simulated from the model.*

The resulting maps from equation (4.28), (4.29) and (4.30) can be seen in Figure 6.10, 6.11 and 6.12 respectively.

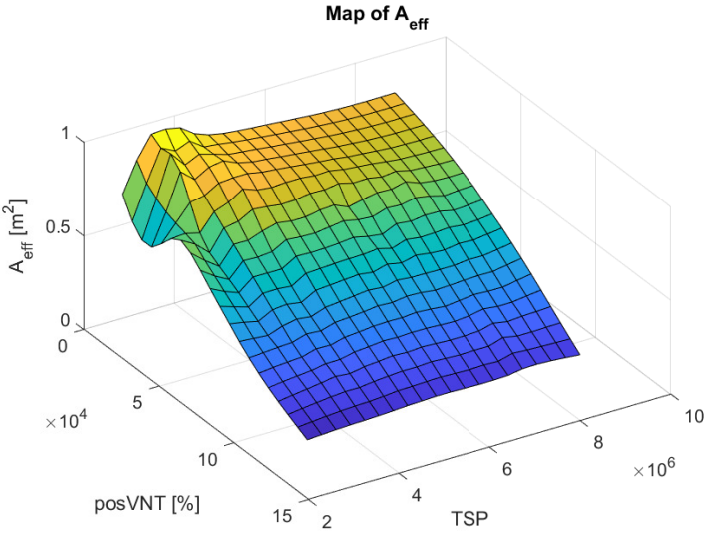


Figure 6.10: *Plot of the map for A_{eff}.*

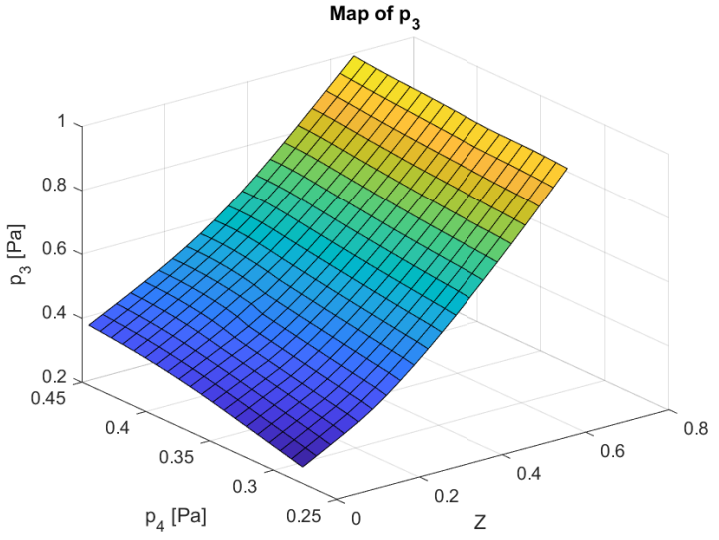


Figure 6.11: *Plot of the map for p_3.*

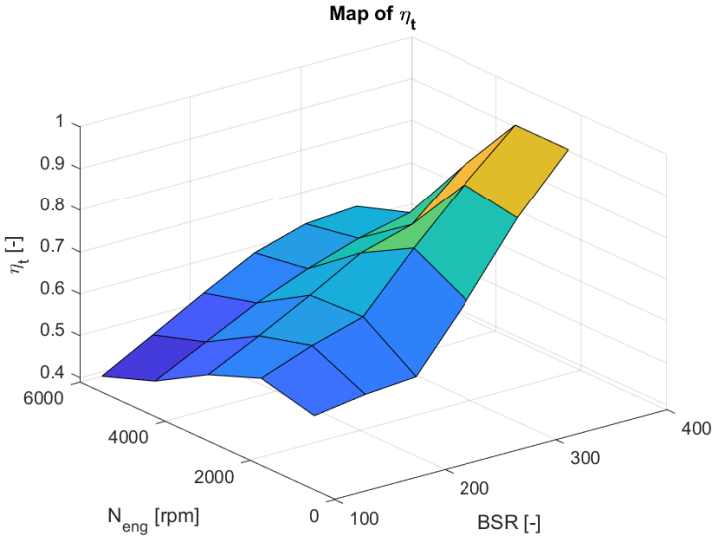
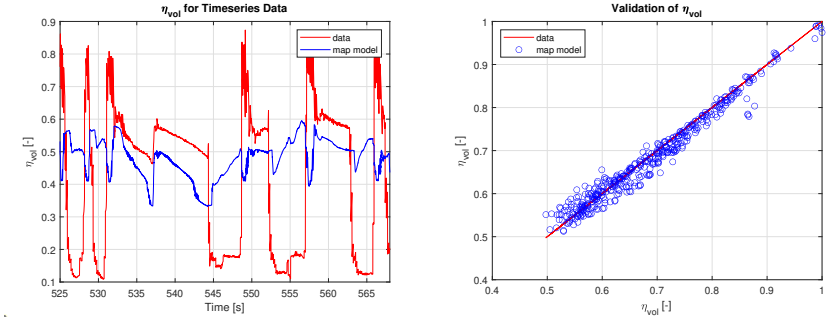


Figure 6.12: *Plot of the map for η_t.*

6.1.3 Cylinders

The cylinder model consists of the equations presented in chapter 4.3. The values from the model for the volumetric efficiency η_{vol} can be seen in Figure 6.13b. The data for the volumetric efficiency comes from the specific cylinder data set that have not been used for any other model. Compared to this data, the model works well with rarely any significant error. However, the quality of the model decreases significantly when used together with the timeseries data as can be seen in Figure 6.13a.



(a) *Plot of η_{vol} for the SIMULINK model and calculated values from the timeseries data.*

(b) *Plot of values for η_{vol} given from the cylinder data set and the simulated values from the model.*

Figure 6.13: *Validation of η_{vol} for the timeseries and volumetric efficiency data sets respectively.*

The model for the air mass flow, $\dot{m}_{air}$, can be seen in Figure 6.14. The model follows the transients in the data but is not able to give as high nor as low values as the data. This model is used to obtain a $\dot{m}_{air}$ that should follow a given reference of the mass flow in the controller for X_{vnt}.

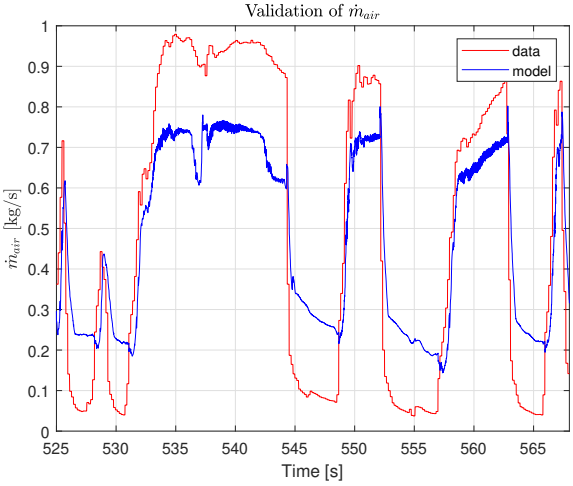


Figure 6.14: *Plot of $\dot{m}_{air}$ given from the timeseries data and simulated from the model.*

The maps used in the cylinder model, representing equation (4.36) and (4.37) can be seen in Figure 6.15 and 6.16 respectively.

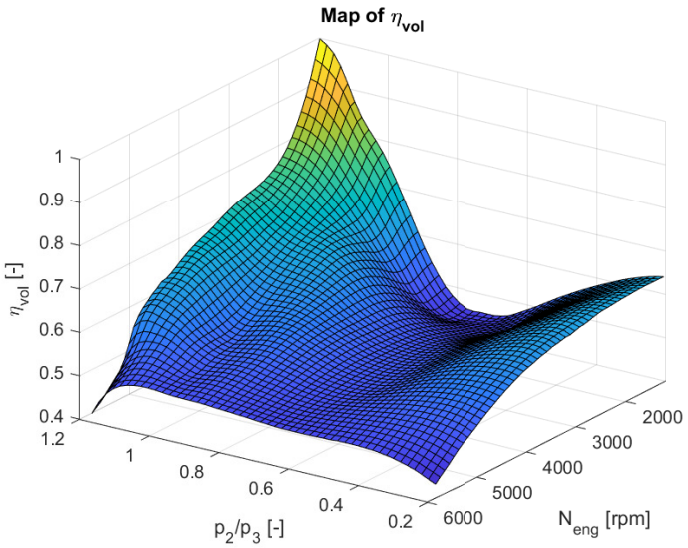


Figure 6.15: *Plot of the map for η_{vol}.*

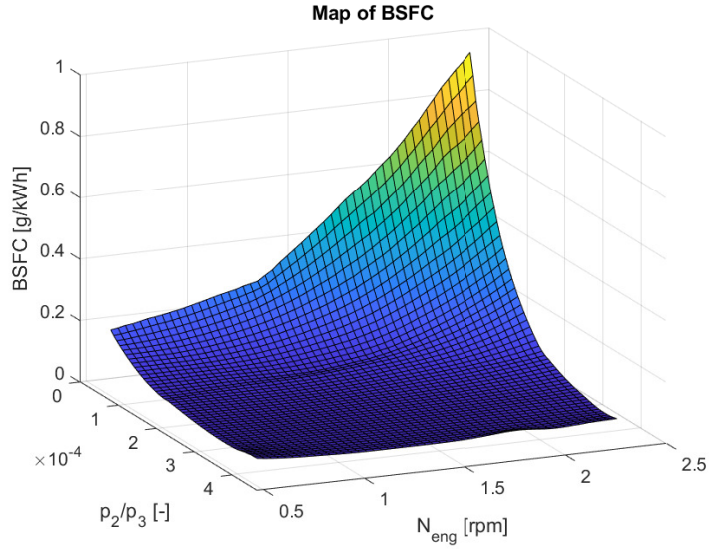


Figure 6.16: *Plot of the map for BSFC.*

6.1.4 Dynamics

The dynamics model consists of the equations presented in chapter 4.4. The final output from these equations are shown in Figure 6.17. All of the previously mentioned equations and models combine to calculate the power generated by the turbine and consumed by the compressor and finally equation (4.40) give the following output for the turbo speed as seen below. The model is capped between a lower and upper limit, thus the abrupt change in derivative at the bottom of the transients. The model is very fast to react to transients, both tip-in and tip-out but does often overshoot its speed in the positive transients where the measured data reaches close to its top speed. The speed also decreases faster for the model than for the data, resulting in lower speed for the model than the measurements in negative transients.

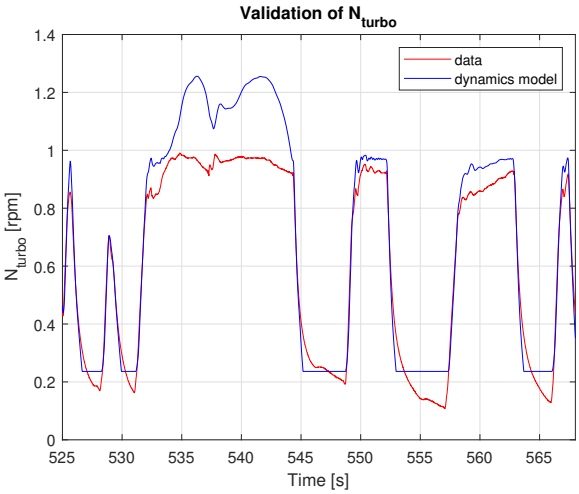


Figure 6.17: *Plot of N_{turbo} given from the timeseries data and simulated from the model. The values have been normalised.*

6.2 Control of the Turbocharger

Below, the results of controlling the model with collocation as well as with a PID are presented.

Due to confidential agreement, all of the values, except for the control signal, presented in the figures have been normalised. The control signal still shows the exact value in order to still understand whether the vanes are actually closed or open. For the other values, this information is not necessary for understanding the results of the report.

6.2.1 Direct Collocation and YOP

The results from trying to solve the OCPs stated in chapter 5.2 are presented here.

The control of the transient 1, which shall follow the mass flow reference and described by equation (5.1), starts at the bottom of the mass flow range and increases to almost reach the highest measured mass flow, can be seen in Figure 6.18. The controller tries to increase the mass flow by closing the vanes of the VNT before it opens up which gives a notable increase in the flow. It is noteworthy that when the vanes are completely open, the mass flow as well as the volumetric efficiency η_{vol} starts to increase. While the controller occasionally manages to be close to the reference, it is quite often a bit off from the reference.

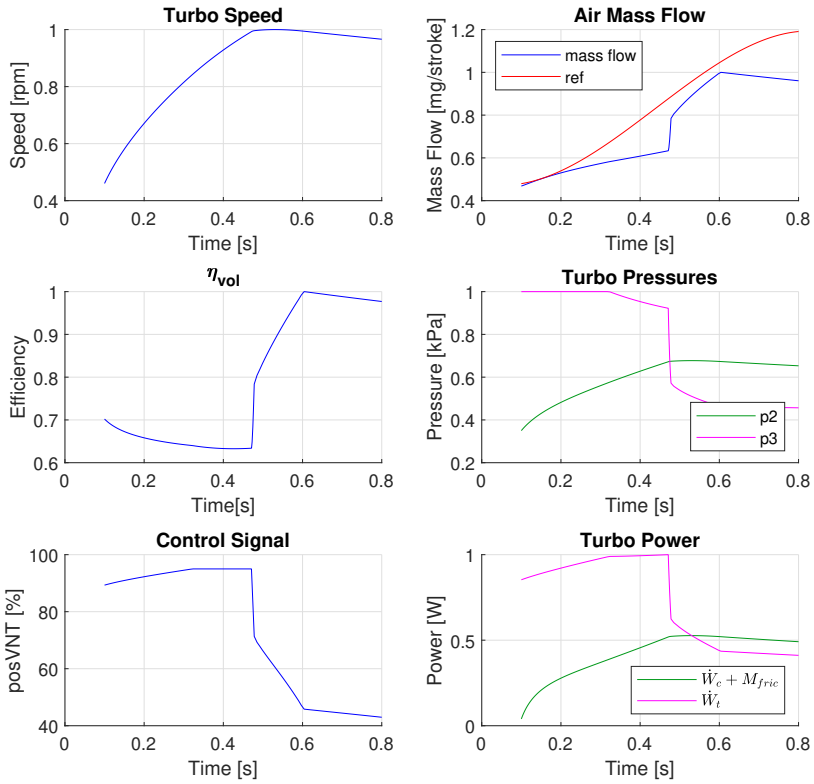


Figure 6.18: *Plots of relevant turbo parameters when solving the OCP for transient 1.*

Transient 2, described by equation (5.2), shall also follow the mass flow reference and starts at the bottom of the mass flow range but then only increases to a value in the middle of the mass flow range. As can be seen in Figure 6.19, the controller is able to keep the mass flow very close to the reference through the entire transient, except for a minor error at the start that probably is due to a poor initial guess. The vanes start at an almost entirely closed state and opens up more during the transient.

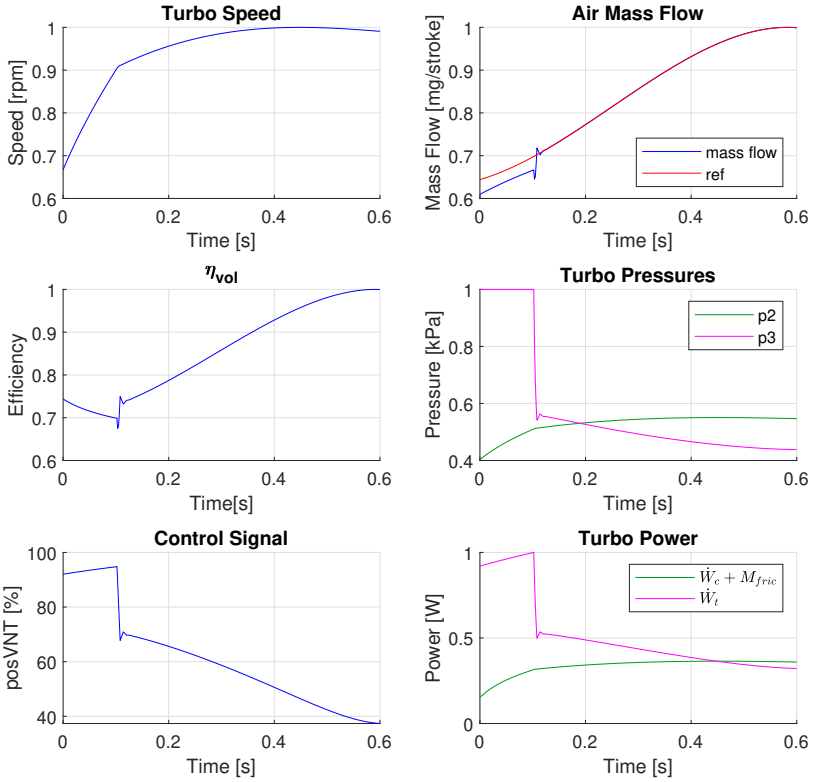


Figure 6.19: *Plots of relevant turbo parameters when solving the OCP for transient 2.*

The third transient, equation (5.3), starts at the middle of the range for the mass flow and tries to follow the mass flow reference to the top of the mass flow range. In Figure 6.20, the mass flow starts at almost the same value as the reference and follows it close to perfect throughout the entire transient. In this transient, the controller chooses to slightly close up the vanes during the early stages and the opens up.

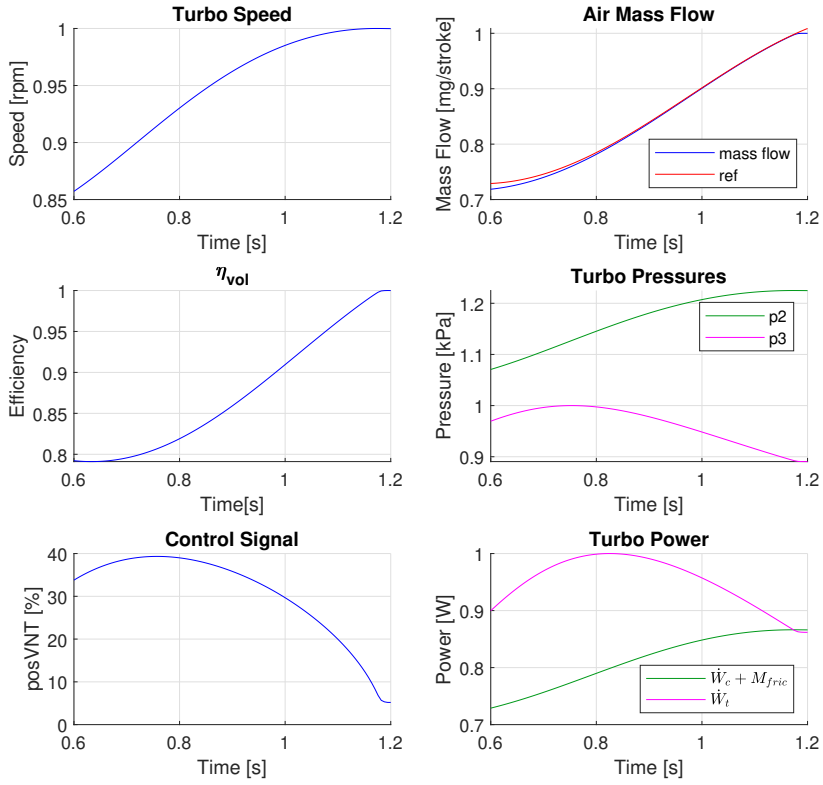


Figure 6.20: *Plots of relevant turbo parameters when solving the OCP for transient 3.*

The result of the "tip-in - tip-out" transient, equation (5.4), can be seen in Figure 6.21. The positive part of the transient in the figure is the same transient as in Figure 6.20. When the controller wants to decrease the mass flow, it does so by further closing the vanes, in this case to maximum closed. This results in an increase of the turbine inlet pressure p_3 which in turn decreases the volumetric efficiency η_{vol}. The mass flow is closely related to the volumetric efficiency and therefore the mass flow also decreases.

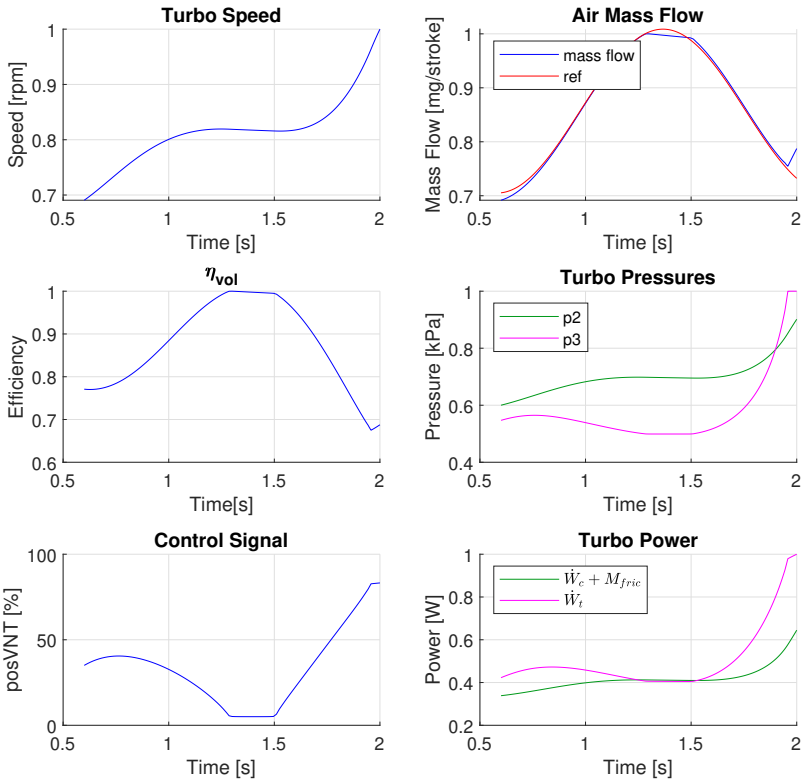


Figure 6.21: *Plots of relevant turbo parameters when solving the OCP for transient "tip-in - tip-out".*

The result for the maximum η_{vol} OCP, equation (5.5) can be seen in Figure 6.22.

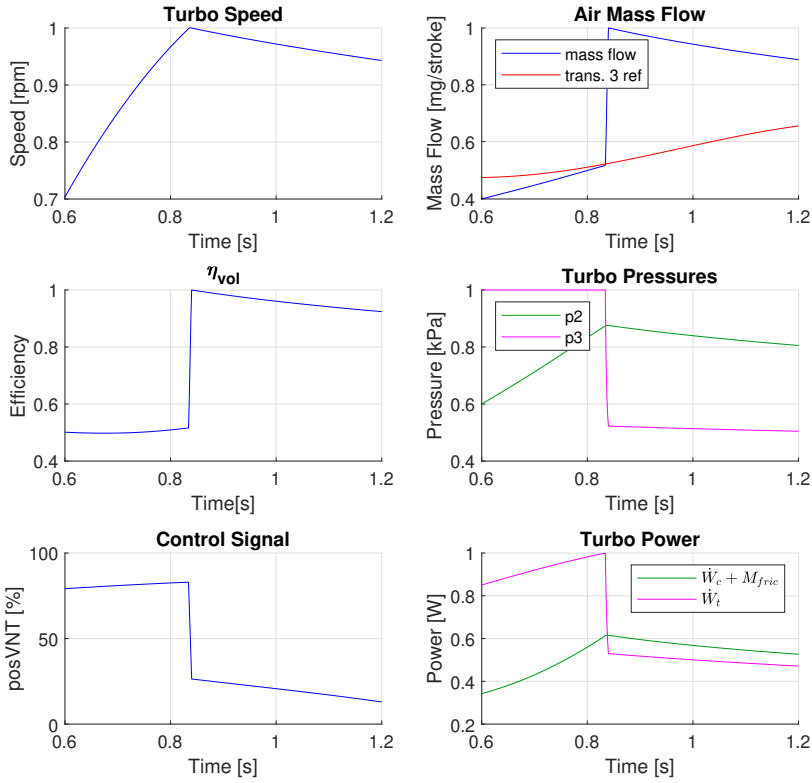


Figure 6.22: *Plots of relevant turbo parameters when solving the OCP for maximum volumetric efficiency.*

The results from solving the OCP for minimising the time taken to reach 0.09 kg/s air mass flow, equation (5.6) can be seen in Figure 6.23. The results from solving the OCP with and without the minimum time constraint, referred to as "constrained" and "free" respectively.

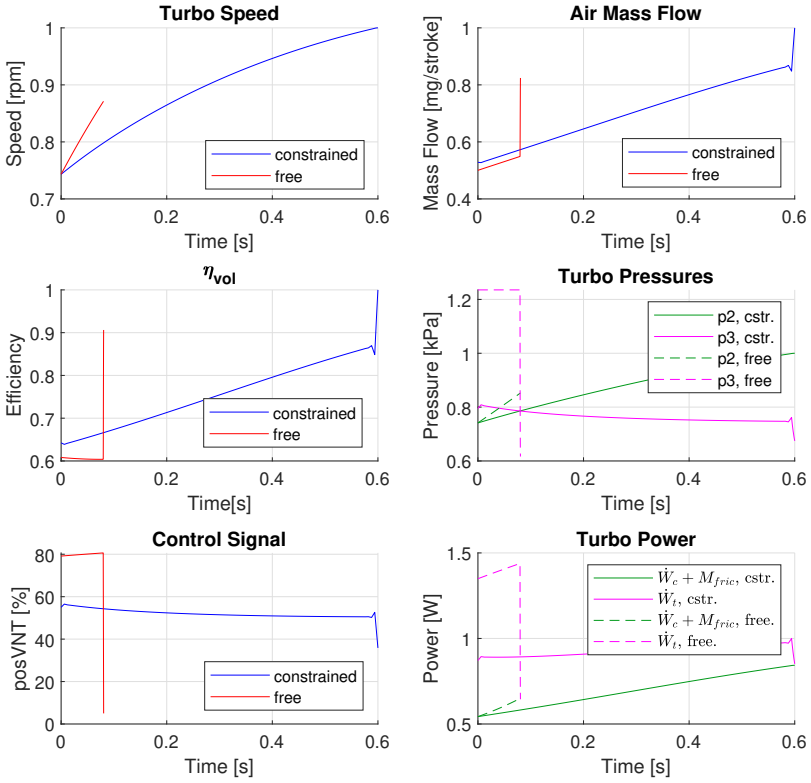


Figure 6.23: *Plots of relevant parameters when solving the OCP for minimising the time. The constrained values represent the OCP for* $0.6 \leq t_f$

6.2.2 PID

In Figure 6.24, the results from the developed PID-controller for transient 3 can be seen. This controller is entirely in SIMULINK. From hand tuning of the PID-gains to make the output follow the reference as good as possible, it was found that $K_p = 750$, $K_i = 2500$ and $K_d = 0$ give decent performance. The tool in MATLAB give approximately the same gains for likewise performance. Since $K_d = 0$, the controller is only a PI-controller. The PI-controller first struggles to follow the steady-state mass flow and is then somewhat late to react to the increase in

the reference. The PI-controller also overshoots the mass flow at the top of the transient.

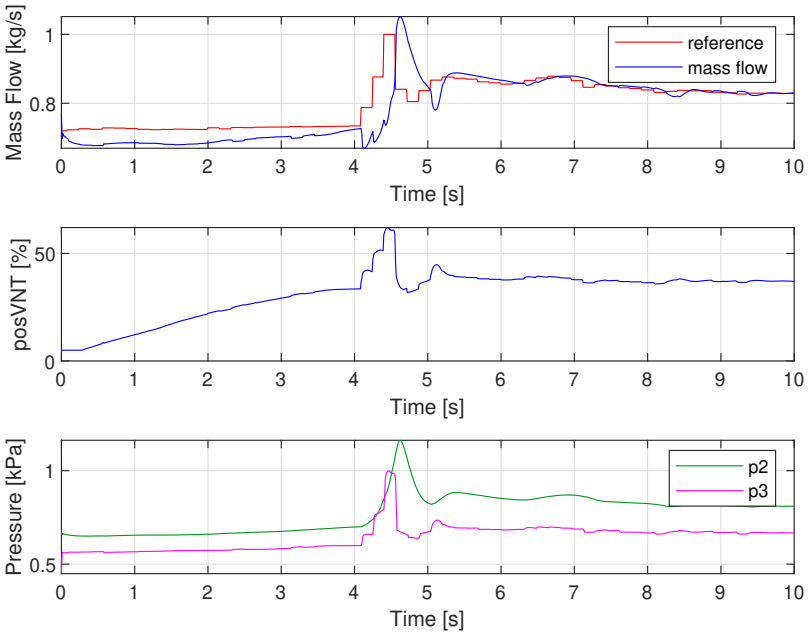


Figure 6.24: *Plot of the air mass flow, control signal, p_2 and p_3 when controlling X_{vnt} to follow a reference for $\dot{m}_{air}$.*

7
Discussion

In this part, the results of the report will be discussed.

7.1 Modelling of the Turbocharger

The maps and models used will be discussed for eventual causes and solutions of problems in this section.

7.1.1 Compressor

The compressor efficiency, η_c, works well when building the model, however, the map appears to be quite inaccurate when simulating the entire model. The map has the corrected compressor mass flow $\dot{m}_{c,corr}$ as one of the inputs. As seen in Figure 6.14, the mass flow has somewhat the same behavior as the efficiency. The corrected mass flow do not differ much from this mass flow and therefore the efficiency might get capped for the same reason as for why the mass flow can not follow the top and bottom of the mass flow from the data. Another explanation could be that the calculated efficiency for the timeseries data actually goes to quite unreasonable values at several occasions where the map generates more reasonable values. This can not be seen in the figure, except for the occasional negative efficiencies, due to normalisation but the value of the peak of the calculated efficiency could be considered unreasonable for an efficiency. Exactly why the calculated efficiency sometimes increases beyond a logical limit is unclear, several of the measured parameters oscillate quite a bit which could add up to unreasonable values. However, the greatest output from the map is in line with the greatest value on the surface of the map, even if the map perhaps should be able to output greater values. Since the map appears to be working well with the

time independent data, together with the arguments mentioned, the map should be considered to be working well when doing simulations.

The pressure ratio, Π_c, of the compressor do not appear to have the same error as the compressor efficiency. This shows that the error in the mass flow is probably not the primary reason for the error in the efficiency. From a visualisation of the map for the pressure ratio, it can be seen that for very high turbo speeds, the mass flow is of less relevance for the output. Since the speed is high where the pressure ratio is big, it can be deducted that the overshoot in the pressure ratio likely is due to the overshoot in the turbo speed in Figure 6.17. p_2 also overshoots its value in the same transients, but since it is directly calculated from the pressure ratio it is most likely not the source of the problem. Correcting the overshoot in the speed would therefore probably also fix the overshoot in the pressure ratio. This overshoot was mainly tuned by adjusting the $K_{inertia}$-term, where an decrease of overshoot occurred at the cost of positive transient response. Alternatively, the map could be adjusted to give lower outputs for higher speeds. In the negative transients, the dynamics of the pressure ratio is very much the same as the dynamics for the turbo speed, probably due to the same reasoning. However, the turbo speed model is capped at a minimum speed which can not be seen to the same extent in the figure for the pressure ratio. From the data the map is built on, a threshold on p_2/p_1 has been applied where the equations used are considered to not be very accurate below this threshold, as previously mentioned in chapter 3. Therefore, this threshold is the lower limit for the pressure ratio in the map. In practice, the pressure ratio rarely goes below this threshold and thus the model appear to give accurate values in this region.

7.1.2 Turbine

Since data to calculate the effective areas of the GPF is missing in the timeseries data set, there is no fair way to validate the model in a simulation of the entire system. From Figure 6.5 and 6.6, it is clear that both tables have a curve where a majority of the points are located, only a few of the data points are not located on this curve. Both tables have some spread data points for lower mass flow, which is not too surprising considering the fact that the entire model loses performance when the turbo is not very active. However, other than a few outliers for the rear area, this spread is not too severe and therefore should not impact the system very much. Since these data points are given, they are considered correct and therefore it is not much to do in order to lower the spread. The model gives the exact same output as the calculations from the time independent data. From these arguments it can be concluded that the tables work properly.

For the effective area of the turbine, A_{eff}, the calculated value from the timeseries data set have not been presented as it only show complex and unreasonably high values. However, the output from the map shows reasonable values. In Fig. 6.7, it shows that the map works very well for all values of the area except for the very highest values where the model becomes somewhat inaccurate, the greatest areas from the model appear to be capped by the maximum values in the map.

It is unclear why the model is worse for higher areas. In contrast to the tables for the areas of the GPF, the value for the area here is calculated from the choke function but with TSP and X_{vnt} as the inputs of the map. Because of this grey box, the values should not be expected to perfectly match as previously. When calculating the area with the time independent data, Ψ is the only value that is not directly from the data. One possible explanation for the inaccuracies could therefore be that the choke function is inaccurate for lower values, as it is reversely proportional to the area.

Because the timeseries data of p_3 is oscillating quite a bit, it is hard to know exactly how accurate the map is. In the negative transients, the map shows some of the same properties as the turbo speed does. Since the pressure indirectly depends on the turbo speed through most of the inputs to the map, it is very likely that they both have this behavior for the same reason. The reason for why the pressure does not get capped like the turbo speed or pressure ratio is likely due to the fact that the pressure has a practical lower limit where it gets increasingly harder to decrease the pressure below this limit. The timeseries data also do not contain values on the edge of the lower limit of the map. For the peaks of the oscillating data, the map most often gives very reasonable values in the middle of the oscillating range.

The map for turbine efficiency, η_t, appears to be very random compared to the time independent data. The reason for why the timeseries data is not used to validate the model is because the calculated efficiency with that data gives a majority of unreasonable values several times higher than what the practical limit of the efficiency would be. The green model in Figure 6.9 is created from the time independent data set and the map contains a lot of unreasonable values for the efficiency. Since the efficiency was calculated from dividing the power of the compressor by turbine power, error in the power calculations could be one reason for why the efficiency model is showing very poor accuracy. Another reason could be the fact that the mechanical efficiency η_m is omitted and assumed to be a part of the turbine efficiency. However, the mechanical efficiency mostly consists of friction losses and should therefore not increase the overall efficiency of the turbine, yet it could explain the poor performance where the map gives reasonable efficiencies. As for the blue model in the figure, since it have not been developed at all during the project, but rather made by AUROBAY, there is not too much to discuss about it. The map does give reasonable values, although not very fitting to the data. It also enables the entire turbocharger model to output a turbo speed with good accuracy. The map has engine speed and BSR as inputs. The engine speed is not something that is calculated in the project and should therefore be considered to be perfect at all times. However, BSR in the simulation is quite often too large compared to what calculations say it should be. From Figure 6.12 it can be seen that a larger BSR always give a larger efficiency which could explain why both models often gives a greater efficiency than the data. The large BSR stem from a saturation used in SIMULINK in order to avoid negative roots. This saturation can be seen in Appendix A.8. To get a better performance from the efficiency map, this saturation should probably be moved or adjusted.

7.1.3 Cylinders

The volumetric efficiency is likely to be the least accurate out of any of the maps, which is quite bad considering it is one of the most central properties of a turbocharged engine. The map has a satisfactory performance when used on the data it is built on. Compared to, for example the given η_t-map in Figure 6.12, the map for volumetric efficiency in Figure 6.15 and the map for BSFC in Figure 6.16 are built on a $50x50$ matrix instead of a $5x5$ matrix. Therefore overfitting could pose a problem and while this could be true for some parts of the map, an overfit that would cause such a bad performance in the simulation would likely be almost perfect when used on the data it is built on. Thus overfitting is likely not the most problematic error in this map. In the map, a big part of the inputs, cylinder pressure ratio and N_{eng}, are actually outside the area of the defined map. For the pressure ratio, a majority of the values from simulating the model are close to or over the upper edge of $p2/p3$ on the map. Some values extend a very significant distance over the edge, these values should very clearly not have the same volumetric efficiency as the map gives. Because these values are capped so far off the map using the clip-function, a big change in the pressure ratio is needed in order for the pressure to have an impact, otherwise the output will only change along the N_{eng} axis. Also, almost half the map is not currently "used" in the simulation as the pressure ratio never gets so low. One reason for the poor fit between map and data could be that the special cylinder data comes from another engine. This engine however does not vary very much from the engine simulated in the project, other than being a bit larger and therefore should not give such poor results. The map is a grey box where the data in the timeseries simulation is calculated from equation (4.31) but the pressure ratio, which is not present in the equation, is used as input. However, this should not be a major concern since the same principle is used for several of the other maps with considerable better performance. It does become harder however to validate the model in the simulation as there is no guaranteed correct volumetric efficiency directly from data. Most likely, the engine difference combined with a very different drive cycle for both sets of data is the greatest source of error for the model. Another problem could be that the map is created from the MATLAB "Curve Fitting Toolbox". The toolbox can give a very good fit but with the risk of having unwanted properties when going outside from where the map is fitted. This behaviour, which will be discussed more in-depth in section 7.2.2, could either help or spoil the performance of the map, but in SIMULINK the map is capped in its output using clip and this problem should therefore not affect the simulation. Since BSFC generally give low fuel consumption where η_{vol} is high, as can be seen in Figure 6.15 and 6.16, a high volumetric efficiency could be considered to give good fuel consumption

The air mass flow $\dot{m}_{air}$ through the cylinders is calculated from equation (4.33) where p_2, η_{vol} and T_2 are the parameters that are calculated within the project. For T_2, in equation (4.6), all inputs have proven to have decent accuracy and therefore T_2 should also be considered to be accurate. Likewise, p_2 comes directly from Π_c and should exhibit the same properties and accuracy. Therefore, the error in the air mass flow should derive almost exclusively from the error in volumetric

efficiency. The mass flow however, shows a great improvement in relative error compared to the volumetric efficiency. This behaviour could potentially come from the fact that the other parameters have a greater impact and therefore saves the model from the poor volumetric efficiency. However, a more likely explanation is that the simulated efficiency is actually better than what Figure 6.13a shows, possibly proving that the equation used to calculate the efficiency data, equation (4.31), would need refinement. The volumetric efficiency would still be bad, judging by the poor mass flow, but not quite as bad as in the figure.

7.1.4 Dynamics

The final output of the model, N_{turbo}, displays some varying performance. The tip-in transients are the most important ones for the project, the accuracy of speed in the lower regions and in tip-out transients are of lesser importance. The model sometimes fails to react to smaller changes in the speed and react very fast when the speed decreases. These fast negative transients are not very interesting and can be neglected. The overshoots of the speed can partly be explained by the $K_{inertia}$ gain for the derivative, the "$-K-$" gain-block visible in the middle of Appendix A.4. This gain have been hand tuned to acquire demanded properties, which in this case is fast and accurate speed in the tip-in transients. By tuning this, the overshoot can be decreased but the current gain appear to give a decent speed with other parts of the model showing decent performance as well. The M_{fric}-term in equation (4.38) is also hand tuned, mainly in order to properly handle steady state in the simulation which it currently does well. Small changes in this term can affect the output quite a bit and could explain some of the overshooting as it is always affecting the model, not only during steady state. The overshoot of around 20% in Figure 6.17 is one of the greatest overshoots and a big majority of the outputs show values with approximately the same overshoot as the other transients in the figure, giving very decent performance for the entire VNT model.

7.2 Control strategy

In this section the control signal for the direct collocation, YOP and the PI controller will be discussed.

7.2.1 Control Signal for Direct Collocation

Regarding the results of the transients, the controller had varying difficulty to minimise the error to the reference. Keep in mind that each OCP is calculated using a fixed engine speed. Transient 1 does not manage find a solution for the given initial turbo value, and the initial N_{turbo} was raised to be above the lower limit to make the problem converge. The solution does not manage to follow the reference nor the amplitude of the reference and this could be blamed on a too low engine speed for such high air flows. Transient 2 performs a good follow after settling in to the reference during the first 0.1 sec. This may be due to poor initial

conditions compared to the reference and therefore need a little time to make up for it. Noticeable is that the OCP needed a constraint on T2 to not "cheat" and start at negative values, and this could be a sign of a flaw in the model for that operation region e.g. by poor fitted functions. Transient 3 have an almost perfect performance, but it struggle to follow for the largest air flow in the top right corner of Figure 6.20. This occurs simultaneous when the control signal hit its lowest value and the model could be considered to have reached its maximum air flow for this operational point. Commonly for the three transients are that the X_{vnt} opens up for an increased volumetric efficiency, and that the pressure ratio $p_2/p_3 > 1$ for higher air mass flows. The volumetric efficiency, and further the mass flow, are the least accurate model in the SIMULINK but the controller still manages to more often than not properly follow the mass flow reference from the volumetric efficiency it is given.

The "tip-in - tip-out" transients as mentioned are a simulation of transient 3, only for a longer duration. And similarly the air-flow at the top does not manage to follow the reference as could be expected by the results from transient 3. However, it may now be easier to see that during the capped air flow, the control signal reaches its bottom resulting in the largest A_{eff} possible, see Figure 6.10. And by studying equation (4.21) and Figure 6.11 one can understand that the p_3 can get capped low, as can be seen in Figure 6.21. Not allowing p_3 to decrease limits the pressure ratio over the cylinder and thus the volumetric efficiency and air mass flow will not increase. The controller manage to follow the tip-out of the transient well. What is interesting is how the controller achieves this. By looking at the control signal it is clear that it closes the vanes during the tip-out and thus the p_3 increases, resulting in a reduced $p2/p3$ and thus a decrease in η_{vol} as can be seen in Figure 6.15 and in agreement with equation (4.33) the air mass flow then decreases. However, this is not the intuitive way of decreasing the air mass flow and it may have some drawbacks. When the p_3 increases we can see how the turbo speed and turbo powers increases, which are not wanted conditions even if lower mass flow were desired, and this can be seen in the end of the simulation where the air mass does not manage to follow and instead starts to increase. The more intuitive way would rather be to open the vanes in order to lower boost pressure and therefore eventually the mass flow.

For the "maximum volumetric efficiency" OCP there is some major differences in the control, compared to transient 3. To start with, the maximum volumetric efficiency increases (about 34%). This is to be expected. The controller utilises the possibility to start at a different operating point which can be seen in Figure 6.22. By closing the vanes and maximise the p_3 the turbo powers and turbo speed increase and builds up a p_2. When the p_2 is sufficient enough to achieve a large pressure ratio as possible, the control signal drops and the vanes open up so p_3 drops resulting in a increased volumetric efficiency. One possible reason for this rather extreme control of the pressures is that the engine speed N_{eng} is constant during each transient. This results in a system where volumetric efficiency is considerably more important in order to control the engine.

The free "minimum time" OCP manage to reach an airflow of 0.09 kg/s in less than 0.09 sec and chooses to solve the constraint of the air mass flow similar to the volumetric efficiency by creating a large pressure ratio enough to generate the η_{vol} needed, see Figure 6.23. Regarding the constrained version, where it must take at least the same amount of time to reach 0.09 kg/s as transient 3, now result in a different control strategy of the vanes by keeping them at almost a constant angle. At a steady pace the turbo powers generate a large enough p_2 and air mass flow. The behaviour in the end is inexplicable currently, likely caused by some constraint within the tool. However, it appears that for the same initial guess as for transient 3, the "minimum time" control strategy can achieve a larger air mass flow and does not get capped.

7.2.2 YOP

Regarding the implementation of YOP, the main concern are the fitting of the maps. Since YOP is not yet able to use the MATLAB interpolation tool "*interp2()*" the maps had to be converted to fitted functions, not always following the intended surface, leaving a source of error if values from that area were to be used. In addition, the fitted functions do not end at the edge of the map but continues further where the maps in SIMULINK have taken advantage of the clip-function. Instead, the fitted function, often a polynomial of a higher degree, continues to move further away from the original data points of the map and into a region where the fitted function thus can behave unpredictable. This was to some extent taken care of by implementing constraints of each parameter earlier represented by a MAP to make sure that no extreme values occurred. The number of collocation points had a great influence on the calculation time, and the more points lead to finer divided problem at the cost of heavier calculation and increased calculation time. The simulations for the transients used 100 control intervals due to a reasonable trade off, where an increase in intervals did not generate a better solution only increased calculation time.

YOP is a new software and with a lack of experience within the project group and a limited amount of time. The implementation of YOP is on a basic level understanding. Dealing with optimal solvers is a delicate task and it will take advantage of a poorly constrained problem or an ill-defined model without hesitation. Thus, why the results of the simulation in the beginning and end behave the way it does is uncertain. For instance how the initial guess affect the resulting solution of the problem, with and without a constraint. Using a constraint for a value at t_0 the simulation starts there, but at an instance continues to the path similar to the solution using no constraints. Some of this due to the fact that the actuator is allowed perfect control and can change value at an instant, as can be seen in the figures of chapter 6.2.1. To properly mimic reality, the actuator should have a low-pass filter applied to it. This would likely remove some of the instant changes in behavior. However, an actuator that can move in an instant could be more interesting to analyse as it would be the truly optimal performance of a perfectly reacting control signal.

7.2.3 PID

The PI gives very poor performance compared to YOP for transient 3. From Figure 6.20 and 6.24 it can be seen that both controllers have roughly the same initial values when the transient begins. Initial values should therefore not be regarded as a major reason for the difference in performance. The control signal for both controllers initially increase in the transient, however, YOP quickly decides to open up the vanes again as a sufficient flow through the engine is achieved while the PI continues to close it during the entire tip-in transient. The fact that the PI gives an overshoot of the mass flow is quite peculiar since the SIMULINK model very rarely gives a mass flow even close to the peaks. There is a possibility that this error in the model causes the PI to over compensate and close the vanes more than needed, eventually resulting in an overshoot. This error can be adjusted by altering the different gains, but a lower K_p or K_i would result in an even slower controller, which is the main problem of this controller. An increase in K_d would result in a more oscillating mass flow. As can be seen by the end of time in Figure 6.24, the resulting mass flow is already on the verge to oscillate and even a slight increase in K_d would quickly cause this controller to lose control. The initial drop in mass flow as the vanes start to close could very likely be a reason for why the controller is behind the reference in time. The drop also increases the error to the reference, forcing the controller to further increase its control signal. The combination of being behind in time as well as an increased error could therefore very likely be the cause of the overshoot in mass flow. One possible explanation for the drop could be that p_3 is faster to react to the change of control signal than p_2, as shows in Figure 6.24. This would lower the volumetric efficiency which could lead to a decrease in mass flow. While this is most often true, it is not always the case as in the middle of the transient. The low mass flow can likely not handle the demand from the control signal properly, which is not a problem with the higher mass flow later in the transient.

8

Future Work

To reduce the source of error in the modelling, the cylinder model could instead be build on maps created by more suitable data. Since the volumetric efficiency is the most uncertain model, focusing on this model would probably yield the most improvement of the model. It would also be interesting to implement control volumes on the cold side to validate if the current assumption of no control volumes due to small volumes is valid. This could be done by examining the time constants of the models with and without the control volumes. $K_{inertia}$ and M_{fric} are currently hand tuned with no mathematical proof of whether or not they are actually close to being optimal. A proper way of tuning these could result in improved performance for the models. The entire engine could be modelled as well, not just the turbocharger, in order to have less simplifications in the models.

Regarding YOP, further experiments to test to solve other optimal control problems in the control strategy, such as a problem containing two objective functions and applying weights to them using a weighted sum approach could be done. This is to study, for instance, a scenario where one objective function wants the transient to go as fast as possible while the other objective function wants to consume the least amount of energy, i.e BSFC. A Pareto front could be used to illustrate the result of such type of OCP. Also, improvements can be done to the fit of the functions in YOP that illustrates the map and a corresponding clip for values that wanders outside the map-region could be implemented. Currently, a constant engine speed is used in YOP. To properly catch the dynamics of a transient, the speed should be able to vary as well. This would possibly change the control signal so that the other parameters are more practically accurate. To compare the performance between a VNT and a FGT the models could be used with data from an engine using FGT and e.g. compare the respective mass flows. The two turbo configurations could also be compared in a real car or test bench

engine allowing a comparison between the acceleration and fuel consumption.

To experimentally validate the controller, the control strategy of a transient could be implemented in an actual vehicle. This would show if the controller works as intended and does not put the engine into an unrealistic or damaging operation area. The results from this experiment could be compared to the performance of the basic controller. In order to test this however, further work is needed to be able to send the YOP-code to the ECU of the engine, alternatively implement YOP in SIMULINK which is easier to use with the ECU.

YOP in its current form is built on CasADi and the IPOPT solver to solve optimal control problems. Different methods have different advantages for different problems and it is possible that using another solver for the NLP or another direct collocation method with other predefined collocation points and a different order of polynomial would result in a different answer. This mathematical part of optimisation in control theory have not been thoroughly investigated in the project and it could potentially be interesting to test the model and the respective objective functions with another solver.

Instead of direct collocation methods, strategies such as MPC could be used. MPC have proved to give good results, as in the litterature study of chapter 1.2.3, and would be a prime target if another strategy should be developed. Focus have been on collocation methods as it is less time consuming and MPC has not been used in this project as it requires a lot of development and could be considering an exaggeration when only having one state and control signal. Collocation methods would also better benefit AUROBAY. However, an MPC would then be well-suited if a control signal for the EGR were to be added. The VNT turbo could also be complemented with an E-turbo, requiring further development to the model to properly include the charge of a battery, further increasing the benefits of using MPC.

9
Conclusion

All of the models, other than turbine efficiency and volumetric efficiency, show very good performance compared to the given data. Despite the errors in the said efficiencies, the model is able to mimic the turbo speed with very good accuracy. The output of the model can be tuned somewhat in order to achieve some different dynamics, it has currently been hand tuned in order to get fast tip-in transients which is the most interesting transient for the project. From this it can be concluded that the model is good in order to analyse the VNT turbo. However, improvement especially in the volumetric efficiency should be a priority in future work in order to be sure about the air mass flow the controller is controlling.

Based on the results from YOP, it can be concluded that direct collocation is a suitable control strategy to control the VNT vanes in a non-linear system. Refinements in the model and the definition of the OCP in YOP would be in place for more reliable and accurate results. Currently, the only transient the controller struggles to follow is where the mass flow goes from its lowest to highest value, which is reasonable as it is the most demanding one. Strategies like MPC and multiple shooting would likely also give equivalent results for the control problem, but would require more development before being useful. A simple PI controller clearly performs worse than the developed controller for the VNT.

The controller chooses to use the control signal very much in the same way as intuition would say. For the three transients where a mass flow reference is to be followed, the controller starts with a closed turbine and keeps it closed until the mass flow is big enough to sustain a bigger turbine. The controller then chooses to open up the vanes, which then also increases the mass flow. For the other objective functions, the controller seems to build up a big flow before it quickly opens the vanes and lets the objective function reach its goal just as quickly. WHen

maximising the volumetric efficiency, the air mass flow increases above the reference for transient 3, showcasing that this could be a viable objective function for achieving fast positive transients. If a varying engine speed would be used, it is very likely that these results would change somewhat, but the results can conclude the logic the controller would use to steer the VNT vanes in a practical application.

Compared to the stopping criterion in section 1.4, it can be concluded that the project has reached its goals and answered the questions at issue. The controller is not yet ready to be implemented in a real engine and can therefore not be seen as a total success. Whether or not the project will be useful for AUROBAY is to be decided in the future.

Appendix

A

Simulink Schemes

Below are schematic figures from SIMULINK of all of the models.

A.1 Compressor

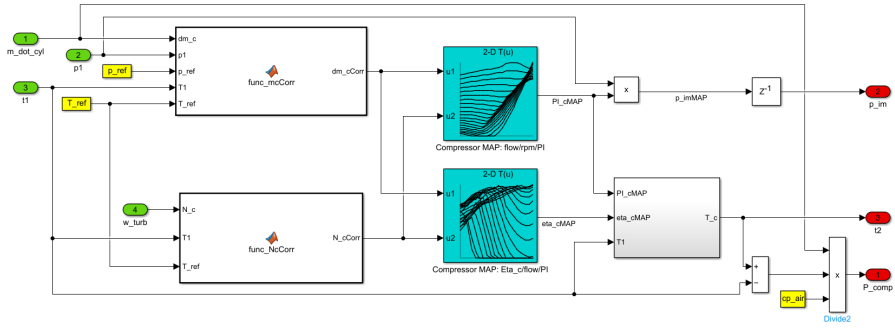


Figure A.1: *SIMULINK scheme of the compressor model.*

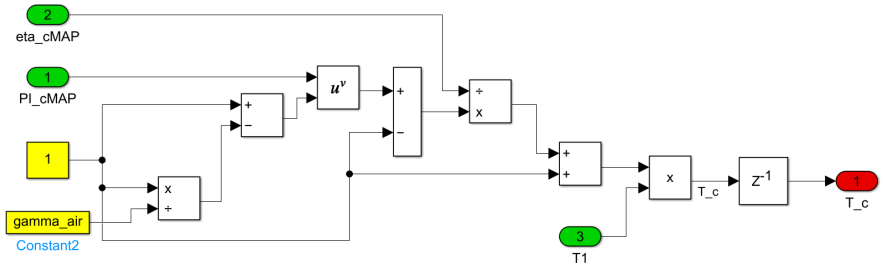


Figure A.2: *SIMULINK scheme of the calculation of T_2 within the compressor model.*

A.2 Turbine

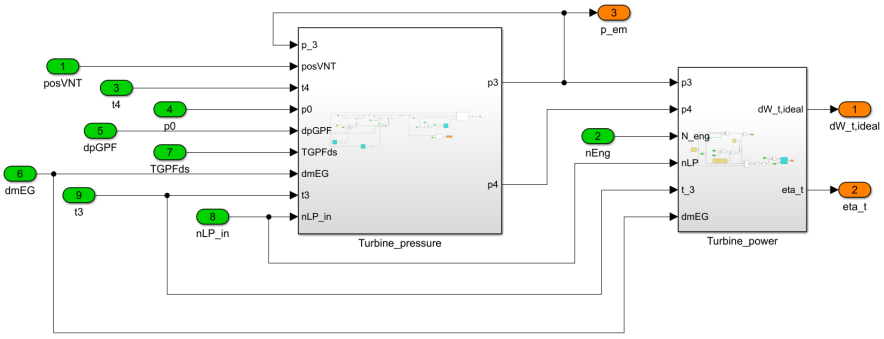


Figure A.3: *SIMULINK scheme of the turbine model.*

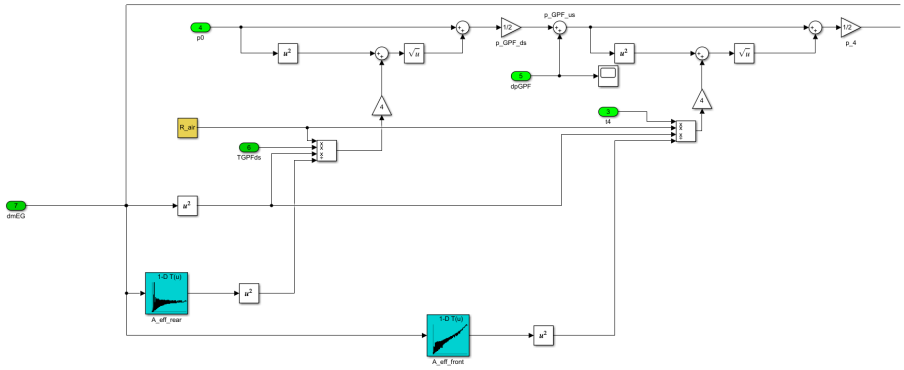


Figure A.4: *SIMULINK scheme of the first half of the turbine pressure subsystem.*

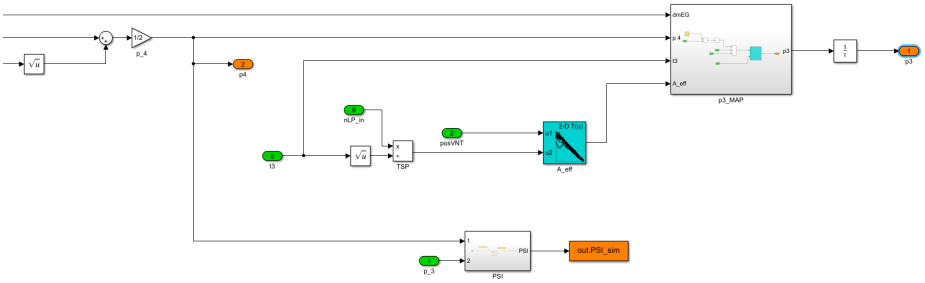


Figure A.5: *SIMULINK scheme of the second half of the turbine pressure subsystem.*

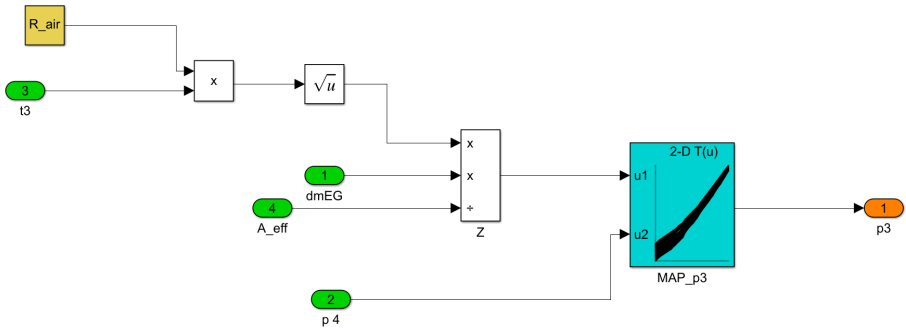


Figure A.6: *SIMULINK scheme of the calculation of p_3 within the turbine pressure subsystem.*

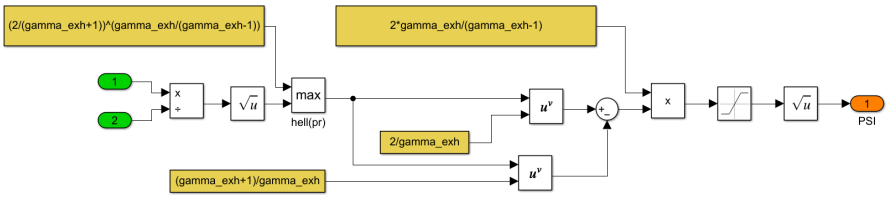


Figure A.7: *SIMULINK scheme of the calculation of Ψ within the turbine pressure subsystem.*

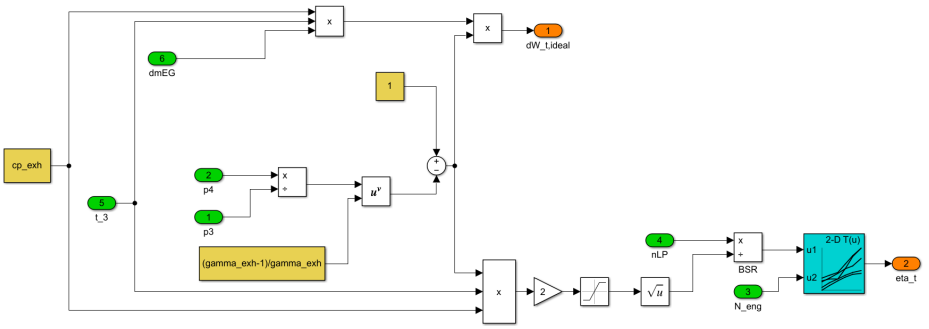


Figure A.8: *SIMULINK scheme of the power and efficiency models within the turbine power subsystem.*

A.3 Cylinders

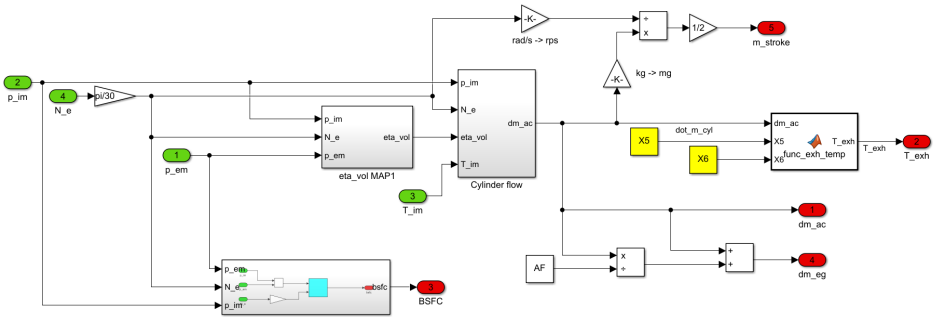


Figure A.9: *SIMULINK scheme of the cylinder model*

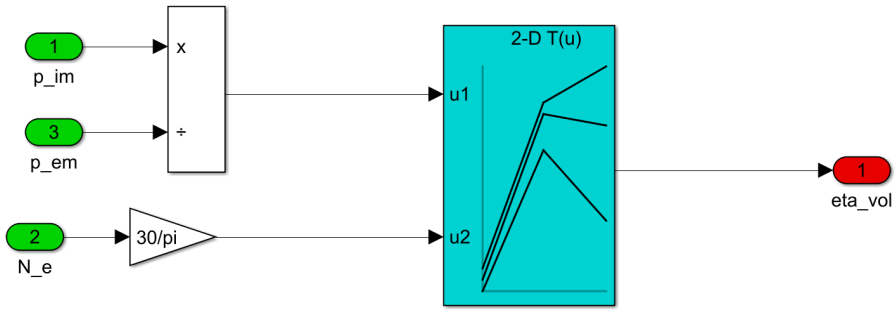


Figure A.10: *SIMULINK scheme of the calculation of η_{vol} withing the cylinder model.*

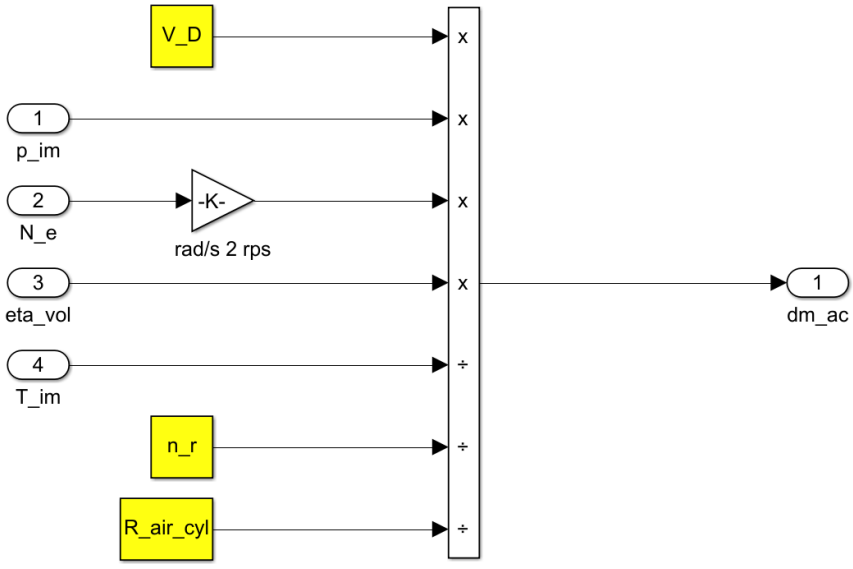


Figure A.11: *SIMULINK scheme of the calculation of $\dot{m}_{air}$ within the cylinder model.*

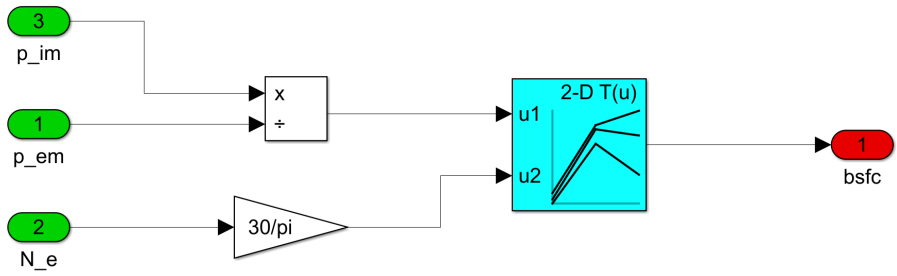


Figure A.12: *SIMULINK scheme of the calculation of BSFC within the cylinder model.*

A.4 Dynamics

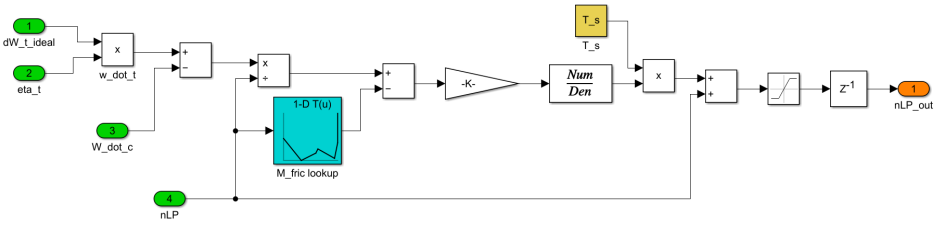


Figure A.13: *SIMULINK scheme of the dynamics model.*

A.5 VNT

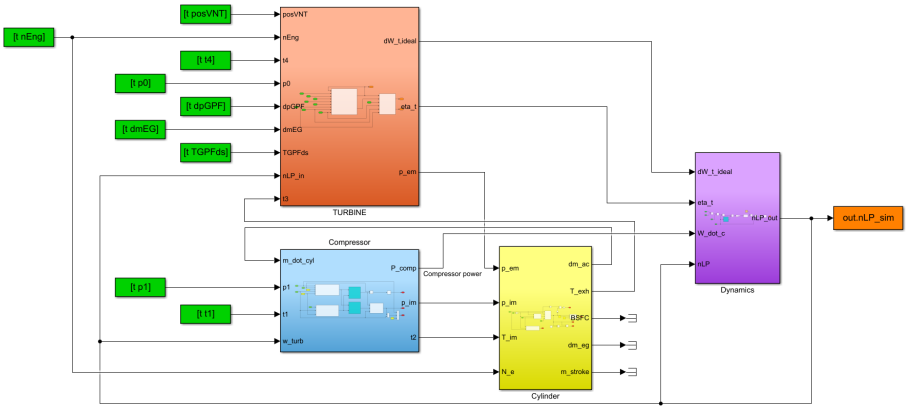


Figure A.14: *SIMULINK scheme of the entire VNT turbocharger model.*

B

YOP Code

Below is the code used to run YOP in MATLAB. Instead of N_{turbo}, the turbo speed is declared as "*nLP*".

B.1 Turbo Control Script

```
%% Optimal Control

% t1 = transient 1, t2 = transient 2, t3 = transient 3

yops Times: t t0 tf %efining independent variable time
yops States: nLP nominal: [134000] %defining state
yops Controls: u_VNT nominal: [28] int: 1 %defining control ...
    signal Piecewise linear by integration.

% States       [rpm]       [kg]
x =     [       nLP     ];
x0    = [       127260  ]; %t1 = 70000, t2 = 76774, t3 = 127260
x_min = [       50000   ];
x_max = [       3e5     ];



% Control    [posVNT]
u =     [      u_VNT  ]; %u_VNT = 0 ==> fully open vanes
u_min = [         5   ];
```

```
u_max = [        95   ];

% Dynamics and outputs
[dx, y] = turbomodel(x, u);

%% Initial guess

%Creating a initial guess simulation environment
sim = yop.ivp( t0 == 0, tf == 2); %start and end time for ...
    initial guess
sim.add( der(x) == dx ); %derivative of state x
sim.add( x(t0) == x0 ); %assign values for the state at t0
sim.add( u_VNT == 24 ); %the control signal. t1 = 10, t2 = 6, t3 = 24

res = sim.solve('solver', 'idas', 'points', 1000); %run the ...
    initial guess simulation

%% OCP

%Creating the OCP
ocp = yop.ocp('Optimal Control');
ocp.min( 100*int( (y.cylinder.dmAir - (0.044 - 0.0394*t + 0.38* ...
    t^2 - 0.288*t^3) )^2)); %transient 1
% ocp.min( 100*int( (y.cylinder.dmAir - (-0.1529*t^3 + 0.1147* ...
    t^2 + 0.02192*t + 0.03884))^2) ); %transient 2
% ocp.min( 100*int( (y.cylinder.dmAir - (0.014*sin(t*4+2.387) + ...
    0.079))^2) ); %transient 3
% ocp.min( tf ); %min time to achieve a certain air mass flow
% ocp.max( int( y.cylinder.eta_vol_fit^2 )); %max volumetric ...
    efficiency to achieve a certain air mass flow

% Problem horizon
ocp.st( t0 == 0.1 ); %OCP min(error), min(t) and max(eta_vol), ...
    t1 = 0.1, t2 = 0, t3 = 0.6, t3_negative = 0.6, ...
    t_min_t = 0, t_max_etavol = 0
ocp.st( tf == 0.8 );  %OCP min(error) and max(eta_vol), ...
    t1 = 0.8, t2 = 0.6, t3 = 1.2, t3_negative = 2, t_max_etavol = 0.6
% ocp.st( 0 <=  tf); %OCP min(t), 0.6 <= tf for constrained


% Differential constraint
ocp.st( der(x) == dx ); %derivative of state x
ocp.st( x(t0) == x0 ); %assign values for the state at t0

% Box contraints
ocp.st( x_min <= x <= x_max ); %state constraint
```

```
ocp.st( u_min <= u <= u_max ); %control constraint

% Terminal and final conditions
% ocp.st( y.cylinder.dmAir(tf) >= 0.09) ; %air mass flow at tf ...
    for min(t), min(t) constrained, max(eta_vol)
% ocp.st( 300 <= y.compressor.T2(t0) ); %OCP min(error) transient 2

% Map constraints
ocp.st( y.turbine.p3_min <= y.turbine.p3 <= ...
    y.turbine.p3_max ); %active for all OCP
ocp.st( y.turbine.eta_t_min <= y.turbine.eta_t_fit <= ...
    y.turbine.eta_t_max ); % OCP min(error) transient 1 and ...
    max(eta_vol)
% ocp.st( y.compressor.eta_c_min <= y.compressor.eta_c_fit ...
    <= y.compressor.eta_c_max );
% ocp.st( y.turbine.A_eff_rear_min <= y.turbine.A_eff_rear_fit ...
    <= y.turbine.A_eff_rear_max );
% ocp.st( y.turbine.A_eff_front_min <= y.turbine.A_eff_front_fit ...
    <= y.turbine.A_eff_front_min );
% ocp.st( y.turbine.A_eff_min <= y.turbine.A_eff_fit ...
    <= y.turbine.A_eff_max );
% ocp.st( y.turbine.M_fric_min <= y.turbine.M_fric_fit <= ...
    y.turbine.M_fric_max );
% ocp.st( y.compressor.PIc_min <= y.compressor.PIc_fit <= ...
    y.compressor.PIc_max );
% ocp.st( y.cylinder.eta_vol_min <= y.cylinder.eta_vol_fit <= ...
    y.cylinder.eta_vol_max );

sol = ocp.solve('ival', 100,'guess', res); %run the ompimization
```

B.2 Turbo Model Script

```
function [dx, y] = turbomodel(state, control)

% states
nLP = state(1);
posVNT = control(1);

% load parameters
turbo_parameter_clean;

% add fit-equations to workspace
```

```
addpath custom_fit
load('A_eff_custom'); load('A_eff_front_custom');
load('A_eff_rear_custom'); load('BSFC_custom');
load('eta_c_custom'); load('eta_t_custom');
load('eta_vol_custom'); load('M_fric_custom_exp');
load('p3_custom'); load('PIc_custom');

%% Compressor

dm_cCorr = func_mcCorr(dmAir, p1, p_ref, T1, T_ref);
N_cCorr = func_NcCorr(nLP, T1, T_ref);
dm_c = dm_cCorr*p1/p_ref/(sqrt(T1/T_ref));
PIc_fit = PI_C_custom.p00 + PI_C_custom.p10*dm_cCorr + ...
    PI_C_custom.p01*N_cCorr + PI_C_custom.p20*dm_cCorr^2 ...
    + PI_C_custom.p11*dm_cCorr*N_cCorr + PI_C_custom.p02* ...
    N_cCorr^2 + PI_C_custom.p30*dm_cCorr^3 ...
    + PI_C_custom.p21*dm_cCorr^2*N_cCorr + PI_C_custom.p12* ...
    dm_cCorr*N_cCorr^2 + PI_C_custom.p03*N_cCorr^3;

p2 = PIc_fit*p1;

eta_c_fit = eta_c_custom.p00 + eta_c_custom.p10*dm_cCorr + ...
    eta_c_custom.p01*N_cCorr + eta_c_custom.p20*dm_cCorr^2 ...
    + eta_c_custom.p11*dm_cCorr*N_cCorr + eta_c_custom.p02* ...
    N_cCorr^2 + eta_c_custom.p30*dm_cCorr^3 ...
    + eta_c_custom.p21*dm_cCorr^2*N_cCorr + ...
    eta_c_custom.p12*dm_cCorr*N_cCorr^2;

T2 = T1*((PIc_fit^((gamma_air-1)/gamma_air) - 1)/eta_c_fit+1);

%% Turbine

T3 = 675.402447247393 + dmAir*13767.5471545029 + ...
    dmAir^2*-184441.451285864 + dmAir^3*1267421.13421623 ...
    + dmAir^4.*-3655341.33164865...
    + dmAir^5*2817242.21886061; %from cylinder temp-fit

dmEG = (dmAir+nLP*0) + (dmAir+nLP*0)/AF; %assume lambda = 1

A_eff_rear_fit = A_eff_rear_custom.a*dmEG^3 + A_eff_rear_custom.b* ...
    dmEG^2 + A_eff_rear_custom.c*dmEG +A_eff_rear_custom.m;

pgpfds = 0.5*(p0 + sqrt(p0.^2+4*dmEG.^2*R_air* ...
    TGPFds/A_eff_rear_fit^2));

pgpfus = pgpfds + dpGPF;
```

```
A_eff_front_fit = A_eff_front_custom.a*dmEG^3 + ...
    A_eff_front_custom.b*dmEG^2 + A_eff_front_custom.c* ...
    dmEG + A_eff_front_custom.m;

TSP = nLP/sqrt(T3);

A_eff_fit = A_eff_custom.m + A_eff_custom.f10*posVNT + ...
    A_eff_custom.f20*posVNT^2 + A_eff_custom.f30*posVNT^3 + ...
    A_eff_custom.f01*TSP + A_eff_custom.f11*posVNT*TSP;

Z = dmEG*sqrt(R_air.*T3)/A_eff_fit;

p4 = 0.5*(pgpfus + sqrt(pgpfus^2+4*dmEG^2*R_air* ...
    t4/A_eff_front_fit^2));

p3 = p3_custom.m + p3_custom.f10*Z + p3_custom.f01*p4 + ...
    p3_custom.f11*Z*p4 + p3_custom.f20*Z^2 + p3_custom.f02*p4^2;

pr_turb = sqrt(p4/p3);

pr_crit = (2/(gamma_exh+1))^(gamma_exh/(gamma_exh-1));
PI = max(pr_turb,pr_crit);
PSI = sqrt(2*gamma_exh/(gamma_exh-1)*(PI^(2/gamma_exh) - ...
PI^((gamma_exh+1)/gamma_exh)));

BSR = nLP/sqrt(2*cp_exh*T3*(1-(p4./p3)^((gamma_exh-1)/gamma_exh)));

eta_t_fit = eta_t_custom.m + eta_t_custom.f10*BSR + ...
    eta_t_custom.f01*nEng/2 + eta_t_custom.f11*BSR*nEng ...
    + eta_t_custom.f20*BSR^2 + eta_t_custom.f02*nEng^2;

%% Turbo power

W_dot_t_ideal = dmEG*cp_exh*T3.*(1-(p4/p3)^ ...
    ((gamma_exh-1)/gamma_exh));
W_dot_t = W_dot_t_ideal*eta_t_fit;

W_dot_c = dmAir*cp_air*(T2-T1)*0.85;

%% Cylinder
PI_cyl = p2/p3;

eta_vol_fit = eta_vol_custom.p00 + eta_vol_custom.p10*nEng + ...
    eta_vol_custom.p01*PI_cyl + eta_vol_custom.p20*nEng^2 + ...
    eta_vol_custom.p11*nEng*PI_cyl + eta_vol_custom.p02* ...
```

```
    PI_cyl^2 + eta_vol_custom.p30*nEng^3 + eta_vol_custom.p21* ...
    nEng^2*PI_cyl + eta_vol_custom.p12*nEng*PI_cyl^2;

dmAir = eta_vol_fit*V_D*nEng*p2/(n_r*R_air*T2*60); %kg/s

%% BSFC

BSFC_fit = BSFC_custom.p00 + BSFC_custom.p10*nEng + ...
    BSFC_custom.p01*PI_cyl + BSFC_custom.p11*nEng* ...
    PI_cyl + BSFC_custom.p02*PI_cyl^2 + BSFC_custom.p13* ...
    nEng*PI_cyl^3 + BSFC_custom.p12*nEng*PI_cyl^2 + ...
    BSFC_custom.p03*PI_cyl^3 + BSFC_custom.p04*PI_cyl^4;

%% Turbo dynamics

M_fric_fit = M_fric_fit.a*exp(M_fric_fit.b*nLP) + ...
M_fric_fit.c*exp(M_fric_fit.d*nLP);

%% Recalculate to get proper variable types

dmEG = (dmAir) + (dmAir)/AF;
A_eff_rear_fit = A_eff_rear_custom.a*dmEG^3 + ...
    A_eff_rear_custom.b*dmEG^2 + A_eff_rear_custom.c* ...
    dmEG +A_eff_rear_custom.m;

pgpfds = 0.5*(p0 + sqrt(p0.^2+4*dmEG.^2*R_air*TGPFds/ ...
    A_eff_rear_fit^2));
pgpfus = pgpfds + dpGPF;

A_eff_front_fit = A_eff_front_custom.a*dmEG^3 + ...
    A_eff_front_custom.b*dmEG^2 + A_eff_front_custom.c* ...
    dmEG + A_eff_front_custom.m;

p4 = 0.5*(pgpfus + sqrt(pgpfus^2+4*dmEG^2*R_air* ...
    t4/A_eff_front_fit^2));
T3 = 675.402447247393 + dmAir*13767.5471545029 +...
    dmAir^2*-184441.451285864 + dmAir^3*1267421.13421623 + ...
    dmAir^4.*-3655341.33164865 + dmAir^5*2817242.21886061;

%% states

%turbospeed derivative
dnLP = K_inertia*(W_dot_t/nLP-W_dot_c/nLP-M_fric_fit);

dx = [dnLP];
```

```
%% Output signals

y.compressor.dm_cCorr =  dm_cCorr;
y.compressor.N_cCorr = N_cCorr;
y.compressor.dm_c = dm_c;
y.compressor.PIc_fit = PIc_fit;
y.compressor.PIc_min = PIc_min;
y.compressor.PIc_max = PIc_max;
y.compressor.p2 = p2;
y.compressor.T2 = T2;
y.compressor.eta_c_fit = eta_c_fit;
y.compressor.eta_c_min = eta_c_min;
y.compressor.eta_c_max = eta_c_max;
y.compressor.T1 = T1;


y.turbine.T3 = T3;
y.turbine.dmEG = dmEG;
y.turbine.A_eff_rear_fit = A_eff_rear_fit;
y.turbine.A_eff_rear_min = A_eff_rear_min;
y.turbine.A_eff_rear_max = A_eff_rear_max;
y.turbine.pgpfds = pgpfds;
y.turbine.pgpfus = pgpfus;
y.turbine.A_eff_front_fit = A_eff_front_fit;
y.turbine.A_eff_front_min = A_eff_front_min;
y.turbine.A_eff_front_max = A_eff_front_max;
y.turbine.TSP = TSP;
y.turbine.A_eff_fit = A_eff_fit;
y.turbine.A_eff_min = A_eff_min;
y.turbine.A_eff_max = A_eff_max;
y.turbine.Z = Z;
y.turbine.p4 = p4;
y.turbine.pr_turb = pr_turb;
y.turbine.PI = PI;
y.turbine.PSI = PSI;
y.turbine.BSR = BSR;
y.turbine.eta_t_fit = eta_t_fit;
y.turbine.eta_t_min = eta_t_min;
y.turbine.eta_t_max = eta_t_max;
y.turbine.nLP = nLP;
y.turbine.posVNT = posVNT;
y.turbine.dmEG = dmEG;
y.turbine.p3 = p3;
y.turbine.p3_max = p3_max;
y.turbine.p3_min = p3_min;
y.turbine.M_fric_fit = M_fric_fit;
```

```
y.turbine.M_fric_min = M_fric_min;
y.turbine.M_fric_max = M_fric_max;

y.power.W_dot_t_ideal = W_dot_t_ideal;
y.power.W_dot_t = W_dot_t;
y.power.W_dot_c = W_dot_c;
y.power.W_comp = y.power.W_dot_c + y.turbine.M_fric_fit*nLP;

y.cylinder.PI_cyl = PI_cyl;
y.cylinder.PI_cyl_max = PI_cyl_max;
y.cylinder.eta_vol_fit = eta_vol_fit;
y.cylinder.eta_vol_min = eta_vol_min;
y.cylinder.eta_vol_max = eta_vol_max;
y.cylinder.dmAir = dmAir;
y.cylinder.BSFC_fit = BSFC_fit;
y.cylinder.nEng = nEng;
end
```

B.3 Turbo Parameter Script

```
%% Turbo parameters

TRANSIENT = 1 %change transient

switch TRANSIENT

    case 1 %300 --> 900 mg/stroke
        dpGPF = 854;
        TGPFds = 1042; % Kelvin
        t4 = 1012;
        p1 = 95491;
        T1 = 302;
        p0 = 97617;
        dmAir = 0.0434;
        nEng = 3847; %Constant


    case 2 %300-->600 mg/stroke
        dpGPF = 1.9743e3;
        TGPFds = 1095.9; %Kelvin
        t4 = 975.54;
        p1 = 9.7642e4;
        T1 = 290.6;
        p0 = 9.7788e4;
```

```
        dmAir = 0.04;
        nEng = 2975; %Constant

    case 3 %600-->900 mg/stroke
        dpGPF = 2758;
        TGPFds = 883; %Kelvin
        t4 = 948;
        p1 = 9.3706e4;
        T1 = 291;
        p0 = 9.757e4;
        dmAir = 0.0684;
        nEng = 3700; %Constant
end

%Constants
gamma_exh = 1.32;
gamma_air = 1.37;
cp_exh = 1183.875;
cp_air = 1062.676;
R_air = 280;
MW = 29; %moluar weight
R_ideal = 8314.5;
AF = 14.7;
T_s = 0.01; %sample time
p_ref = 101.53e3;
T_ref = 298.15;
n_r = 2;
n_cyl = 4;
r_c = 11.25;
V_D = 1.969e-3;
V_d = V_D/4;
V_c = V_d / (r_c-1);
R_exh = 280;
PSI = 0.465;
K_inertia = 5760000;

% max/min map values
p3_min = 0.68687e5;
p3_max = 2.7957e5;
eta_c_min = 0.05;
eta_c_max = 1;
eta_t_min = 0.3888;
eta_t_max = 1;
A_eff_rear_min = 7.8e-4;
A_eff_rear_max = 5e-3;
A_eff_front_min = 8.7e-4;
```

```
A_eff_front_max = 3.331e-3;
A_eff_min = 1.1344e-4;
A_eff_max = 7.2973e-4;
M_fric_min = -9e-4;
M_fric_max = 0.015;
PIc_min = 1.05;
PIc_max = 3.7117;
PI_cyl_max = 1.4;
eta_vol_min = 0.4162;
eta_vol_max = 1.1;
```

C

Fitted Models

The black dots of the figures are the same data the maps are built on, the surface is the fitted function used in YOP. The values have been removed due to confidentiality.

C.1 Compressor

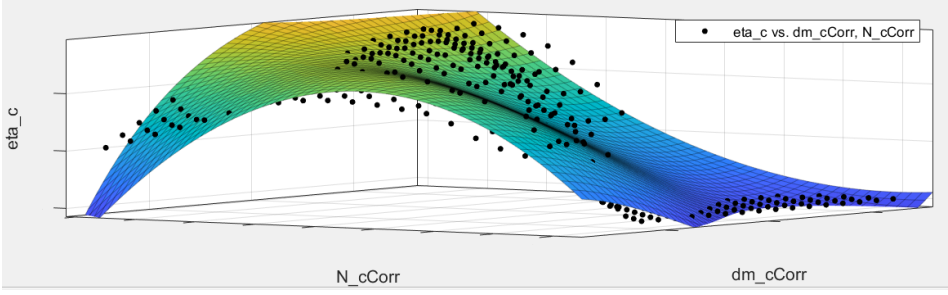


Figure C.1: *Surface of the fit of η_c to the data points of the map.*

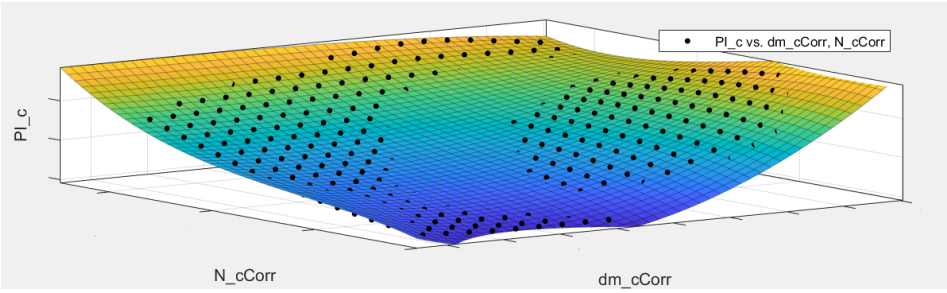


Figure C.2: *Surface of the fit of Π_c to the data points of the map.*

C.2 Turbine

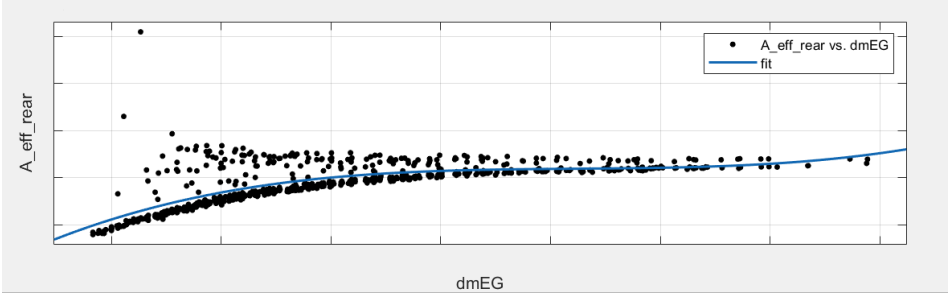


Figure C.3: *Plot of the fit of $A_{eff,rear}$ to the data points of the table.*

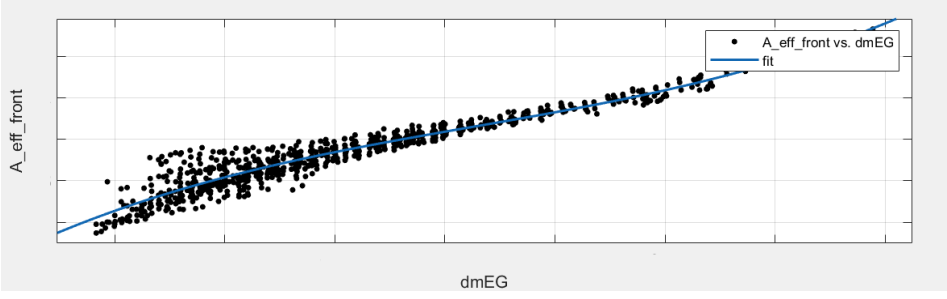


Figure C.4: *Plot of the fit of $A_{eff,front}$ to the data points of the table.*

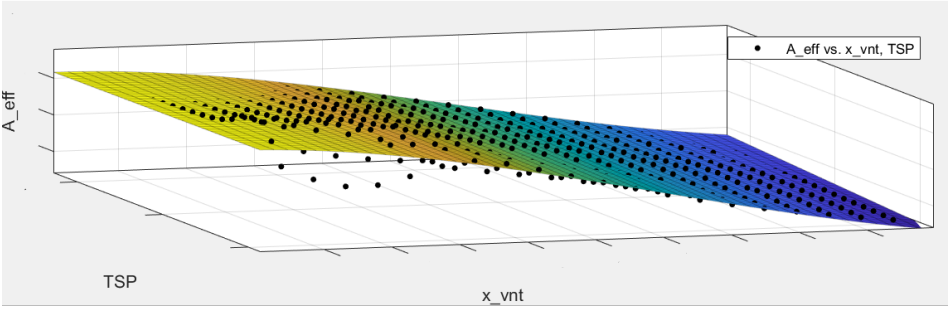


Figure C.5: *Surface of the fit of A_{eff} to the data points of the map.*

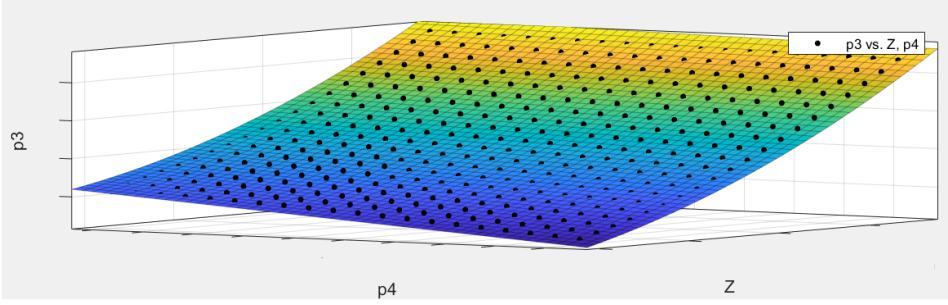


Figure C.6: *Surface of the fit of p_3 to the data points of the map.*

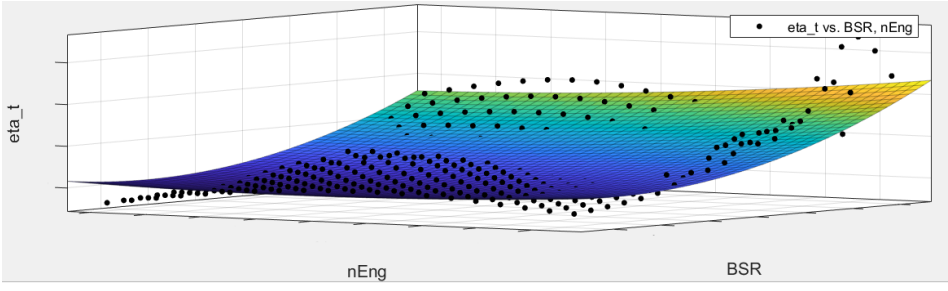


Figure C.7: *Surface of the fit of η_t to the data points of the map.*

C.3 Cylinders

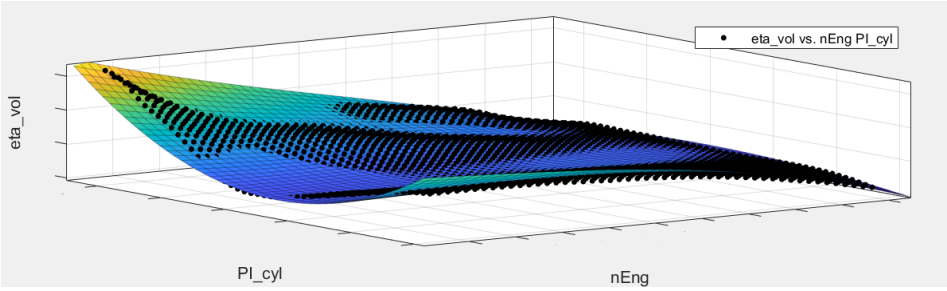


Figure C.8: *Surface of the fit of η_{vol} to the data points of the map.*

C.4 Dynamics

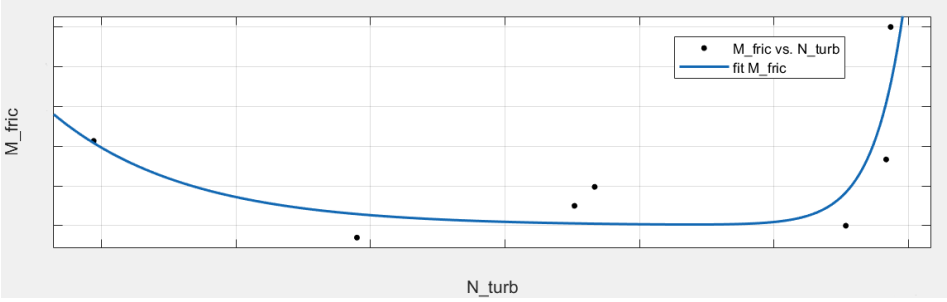


Figure C.9: *Plot of the fit of M_{fric} to the data points of the map.*

References

[1] D Han, S Han, B Han, and W Kim. Development of 2.0l turbocharged disi engine for downsizing application. *SAE Technical Paper 2007-01-0259*, 2007. doi: 10.4271/2007-01-0259.

[2] Adam J. Fenley, Apostolos Pesiridis, and Amin Mahmoudzadeh Andwari. Variable Geometry Turbocharger Technologies for Exhaust Energy Recovery and Boosting-A Review. *Renewable and Sustainable Energy Reviews*, pages 959–975, 2017. doi: https://doi.org/10.1016/j.rser.2016.12.125.

[3] Dezong Zhao, Edward Winward, Zhijia Yang, Richard Stobart, and Thomas Steffen. Robust control of electrified turbocharged diesel engines. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 734–739, 2016. doi: 10.1109/CDC.2016.7798355.

[4] Oscar Flärdh, Gustav Ericsson, Erik Klingborg, and Jonas Mårtensson. Optimal air path control during load transients on a spark ignited engine with variable geometry turbine and variable valve timing. *IEEE Transactions on Control Systems Technology*, 22(1):83–93, 2014. doi: 10.1109/TCST.2013.2245330.

[5] José Ramón Serrano, Pedro Piqueras, Joaquín De la Morena, Alejandro Gómez-Vilanova, and Stéphane Guilain. Methodological analysis of variable geometry turbine technology impact on the performance of highly downsized spark-ignition engines. *Energy*, 215(PB), 2021. doi: 10.1016/j.energy.2020.119.

[6] L Lezhnev, I Kolmanovsky, and J Buckland. Boosted gasoline direct injection engines: Comparison of throttle and vgt controllers for homogeneous charge operation. *SAE Technical Paper 2002-01-0709*, 215, 2002. doi: https://doi.org/10.4271/2002-01-0709.

[7] Lars Eriksson, Lars Nielsen, Jan Brugård, Johan Bergström, Fredrik Pettersson, and Per Andersson. Modeling of a turbocharged si engine. *Annual Reviews in Control*, 26(1):129–137, 2002. ISSN 1367-5788. doi: https://doi.org/10.1016/S1367-5788(02)80022-0.

[8] 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*, 225(7):960–986, 2011. doi: 10.1177/0954407011398177.

[9] I Bahiuddin, S Mazlan, F Imadubbin, and Ubaidillah. A new control-oriented transient model of variable geometry turbocharger. *Energy*, 125: 297–312, 2017. doi: https://doi.org/10.1016/J.ENERGY.2017.02.123.

[10] G Zhu, Gouming and Zeng, Tao. Control-oriented turbine power model for a variable-geometry turbocharger. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 232(4):466–481, 2018. doi: 10.1177/0954407017702996.

[11] J Gonzales Pulpeiro and C Hall. Equation-Based Compressor and Turbine Modeling for Variable Geometry Turbochargers. *SAE Technical Paper 2018-01-0966*, 2018. doi: 10.4271/2018-01-0966.

[12] Serrano J, Arnau F, Dolz V, Andrés T, and Cervelló C. A model of turbocharger radial turbines appropriate to be used in zero- and one-dimensional gas dynamics codes for internal combustion engines modelling. *Energy Conversion and Management*, 49:3729–3745, 12 2008. doi: 10.1016/j.enconman.2008.06.031.

[13] Das H and Dhinagar S. Airpath Modelling and Control for a Turbocharged Diesel Engine. *SAE Technical Paper 2008-01-0999*, 01 2008. doi: https://doi.org/10.4271/2008-01-0999.

[14] Stephan Hennings Och, Luís Mauro Moura, Viviana Cocco Mariani, Leandro dos Santos Coelho, José Antonio Velásquez, and Eric Domingues. Volumetric efficiency optimization of a single-cylinder D.I. diesel engine using differential evolution algorithm. *Applied Thermal Engineering*, 108:660–669, 2016. doi: https://doi.org/10.1016/J.APPLTHERMALENG.2016.07.042.

[15] A Ohata and Y Ishida. Dynamic Inlet Pressure and Volumetric Efficiency of Four Cycle Four Cylinder Engine. *SAE Technical Paper 820407*, 1982. doi: https://doi.org/10.4271/820407.

[16] N. Watson and K. Banisoleiman. A Variable–Geometry Turbocharger Control System for High Output Diesel Engines. *SAE Transactions, Section 6: JOURNAL OF ENGINES*, 97:152–167, 1988. URL `https://www.jstor.org/stable/44547364`.

[17] Huayin Tang, Colin Copeland, Sam Akehurst, Chris Brace, Peter Davies, Ludek Pohorelsky, Les Smith, and Geoff Capon. A novel predictive semi-physical feed-forward turbocharging system transient control strategy based on mean-value turbocharger model. *International J of Engine Research*, 18(10):765–775, 2017. doi: 10.1177/1468087416670052.

[18] Johan Wahlström, Lars Eriksson, and Lars Nielsen. EGR-VGT Control

and Tuning for Pumping Work Minimization and Emission Control. *IEEE Transactions on Control Systems Technology*, 18(4):993–1003, 2010. doi: 10.1109/TCST.2009.2031473.

[19] Shashidhar S Gokhale, Yathisha L, and S Patil Kulkarni. LQR Based Optimal Control Techniques as Applied to Air Path of Diesel Engines. *Proceedings of the Second International Conference on Emerging Trends in Science & Technologies For Engineering Systems (ICETSE-2019)*, 2019. doi: http://dx.doi.org/10.2139/ssrn.3507400.

[20] Johan Wahlström. Modellbaserad prediktiv reglering av en dieselmotor med variabel geometriturbin och återcirkulering av avgaser. Master's thesis, Linköping University, 2003.

[21] Anders Ekdahl. Transient control of variable geometry turbine on heavy duty diesel engines. In *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.*, pages 1228–1233, 2005. doi: 10.1109/CCA.2005.1507299.

[22] Mariagrazia Dotoli and Paolo Lino. Fuzzy adaptive control of a variable geometry turbocharged diesel engine. *IEEE International Symposium on Industrial Electronics*, 4:1295 – 1300 vol.4, 02 2002. doi: 10.1109/ISIE.2002.1025977.

[23] Jonatan Gustafsson. Linearization Based Model Predictive Control of a Diesel Engine with Exhaust Gas Recirculation and Variable-Geometry Turbocharger. Master's thesis, Linköping University, 2021.

[24] Matthew Kelly. An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Rev.*, 59:849–904, 2017.

[25] Xavier Llamas and Lars Eriksson. Optimal Transient Control of a Heavy Duty Diesel Engine with EGR and VGT. In *Proceedings of the 19th IFAC World Congress*, pages 11854–11859, Cape Town, South Africa, 2014. IFAC Papers Online.

[26] Rabih Omran, Rafic Younes, and Jean-Claude Champoussin. Optimal control of a variable geometry turbocharged diesel engine using neural networks: Applications on the etc test cycle. *IEEE Transactions on Control Systems Technology*, 17(2):380–393, 2009. doi: 10.1109/TCST.2008.2001049.

[27] Lars Eriksson Lars Nielsen. *Modeling and Control of Engines and Drivelines*. International series of monographs on physics. John Wiley & Sons Inc, 2014. ISBN 9781118479995.

[28] Robin Holmbom. *Modeling and Model-based Control of Automotive Air Paths*. PhD thesis, Linköping University, 2022.

[29] Viktor Leek. An Optimal Control Toolbox for MATLAB® Based on CasADi. Master's thesis, Linköping University, 2016.

[30] Black box. *Wikipedia*. Accessed May 23, 2022 [Online]. URL https://en.wikipedia.org/wiki/Black_box.

[31] Grey box model. *Wikipedia*. Accessed May 23, 2022 [Online]. URL https://en.wikipedia.org/wiki/Grey_box_model.

[32] Moritz Diehl. Numerical optimal control, 07 2011.

[33] Joel Andersson. A general-purpose software framework for dynamic optimization. *Faculty of Engineering Science*, page 15, 2013.

[34] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, In Press, 2018.

[35] *MATLAB version 9.11.0.1873467 (R2021b) Update 3*. The Mathworks, Inc., Natick, Massachusetts, 2021.

[36] Gabriel Ehlin and Erik Vandor. Eletrically Assisted Turbocharger: Modelling and Control Strategy. Master's thesis, Linköping University, 2022.