CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF MECHANICAL ENGINEERING

DIPLOMA THESIS

MODELLING,SIMULATION AND VALIDATION OF ENGINE MODELS FOR IMPLEMENTATION OF MODEL PREDICTIVE CONTROL ON SI ENGINE AIRPATH

ARULKUMARAN MATHIVANAN

Annotation

Author: Arulkumaran Mathivanan

Title : Modelling, Simulation and Validation of Engine models for Implementation of Model Predictive Control on SI engine airpath

Academic year: 2019/2020

Programme: Master of Automotive Engineering

Major: Advanced Powertrains

Department: Department of Automotive, Combustion Engine and Railway Engineering

Supervisor: Dr .Ing. Jan Macek, Ph.D.

Abstract: This thesis deals with the creation of Engine models to simulate the engine airpath behaviour for implementation of a Model predictive control system to control the engine airpath. The models are simulated and validated against the engine for the above mentioned objective.

Number of pages: 84

Number of pictures: 70

Number of tables: 5

Acknowledgement:

First and foremost I would like to thank each and every being who have contributed to the advancements in science and Technology through history, who all have influenced me directly or indirectly to kindle my passion in the field of Engineering.

I convey my gratitude to Dr.Ing.Gabriela Achtenova and all my professors who have imparted their knowledge and experience in me through my masters course which helped me to write this thesis.

My deepest thanks to Prof. Dr.Ing.Jan Macek who spared his valuable time to supervise my thesis and provide me with guidance and support throughout the activity.

My heartful thanks to Toyota motors Europe for offering me this invaluable opportunity for my internship and my colleagues and team manager Francois Alexandre Lofassas who were always ready to support me in times of difficulties. Particularly I would like to thank my mentor and technical senior manager Shota Nagano for his feedbacks and reviews which helped me immensely. At last and most importantly my greatest thanks to my colleagues Dr.Nicola Pompini and Dr.Remi Losero who were always ready to help me and support me through the activity with their knowledge.

Finally my thanks to all my family and friends who were of great support to me and provided me with the encouragement to complete my thesis.

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publication used in accordance with the methodological guidelines about adhering to ethical principles in the preparation of the final thesis.

Arulkumaran Mathivanan

Contents

List of Figures

List of Tables:

1. Introduction:

The demand for better transportation has been existing since humankind started to civilize. From the use of animals for faster transportation to the high speed machines and the rockets used to break out of our atmosphere we have come a long way. Road transportation has encountered a rapid advancement in the last 100 years since the invention of heat engines and eventually internal combustion engines which are more compact and powerful. Ever since the first internal combustion engine was developed by Étienne Lenoir and the modern petrol engine developed by Nikolaus Otto in 1876 the world has seen an overwhelming advancement in the transportation field.

Today technological growth in the automotive sector is driven by safety, efficiency and comfort. To meet all these requirements automotive companies spend millions of euros in research and development of new technology every year. One of the major factors that had led the automotive sector to meet the technological demands is the successful integration of advanced control systems in the car. From the most common ABS braking system to automatically drivable cars, control systems play a major role in the automotive sector.

The objective of an control system is to basically achieve the desired output by controlling the right amount of inputs by analysing the feedback from the system. A very simple example of a control system is the human brain. When you are on the street driving and you see the red light you automatically press the brake to bring the car to a halt just close to the braking line. So you use an input which is the red light and get the feedback from the car which is the vehicle speed and you calculate the braking input you have to provide the car within the red light. Control systems perform the exact same operations with the help of a computer for the calculation and logic implementation , sensors for system feedback and the actuators for the control.

Today control systems have been widely implemented into modern internal combustion engines. Engines today have to meet the demand for lower emissions and better efficiency to save the environment and reduce the consumption of resources. To achieve these goals many technologies have been implemented in modern engines, such as direct injection, variable valve timing, variable spark timing , turbocharging , EGR etc. It is reasonable to put forth the argument that many of these technologies have been around for a while now. Of course it’s true, but the implementation of better control systems has made it possible to use these technologies effectively for achieving the maximum possible efficiency in engines.[1]

Since the topic being dealt here deals with the implementation of controller on a turbocharged engine airpath it is important to know the main components of an airpath before we understand the need for a controller on the airpath. The engine airpath is basically the flow direction of the air through an engine. In an turbocharged SI-Direct injection engine the air flows through the airfilter, compressor(to pressurize the air), cooler ,Throttle(to control the flow) , Cylinder(for combustion of fuel) , turbine(to drive the compressor) ,catalyst(emission control) and ultimately back into the atmosphere. Here the compressor and turbine work together to utilize the energy from the exhaust gas to pressurize the intake air to increase the volumetric efficiency of the cylinder. This pressure generated at the intake is called the boost pressure and is controlled by adjusting a wastegate valve across the turbine to bypass the exhaust gases flowing over the turbine. This is shown in the figure:

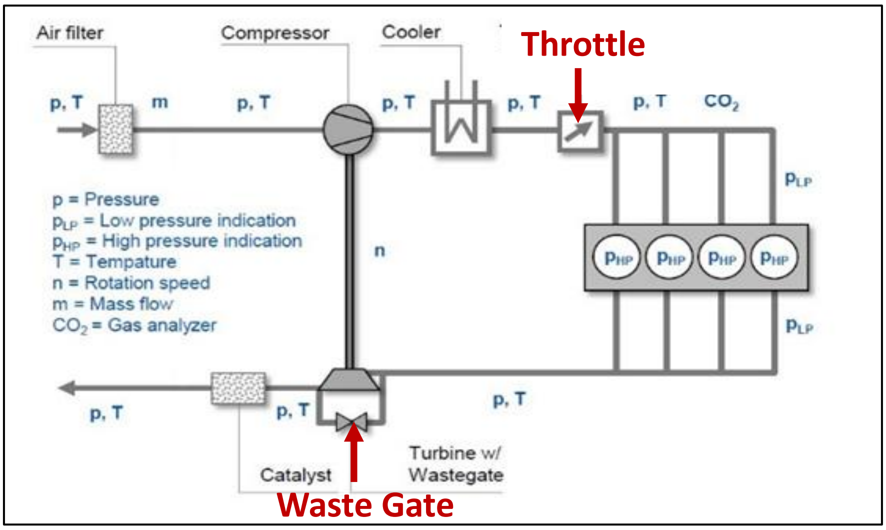


Figure 1: Turbocharged Engine Airpath

As the turbocharged engine uses two actuators to control the airpath of the engine, it now presents us with a problem of controlling both the actuators simultaneously to achieve the right performance from the Engine. A Model predictive control(MPC) system can be implemented to control the actuators. The demand for the implementation of the MPC in the airpath is created by the high costs and time required to calibrate both the controllers simultaneously. The main advantage of an MPC is its superiority in the application of systems with multiple manipulated and control variables[3]. Our case is one such example where the output of the engine being dependant on the pressure and the flow in the airpath which are in turn dependant on both the actuators(throttle and wastegate). This explains the motivation to consider an MPC controller for airpath control.

2. Project Overview:

The objective of this section is to provide the reader with a clear understanding of the project. This section starts with the introduction to Model predictive control as it is important to know the working of the control process before going into the details of the thesis objective. It explains the working of MPC, use of models in the MPC, Development and implementation process of a MPC , the scope of the thesis and workflow of the thesis activity.

2.1. Model Predictive Control:

As the name suggests it is a control system which predicts the future output of an system based on the system model and optimizes the control inputs to achieve the desired output. It chooses the control action by solving an optimal control problem to minimize a performance criterion over a finite future time horizon subjected to constraints. The future behaviour of the system is predicted by the system model used.

$$u_{optimal} = \arg\min_{u} \left(\overbrace{\sum_{k=0}^{Np} \left(y_{target} - y_k\right)^T Q_y \left(y_{target} - y_k\right)}^{\text{Deviation from target}} + \overbrace{\sum_{k=0}^{Np} \Delta u_k{}^T Q_{\Delta u} \Delta u_k + \sum_{k=0}^{Np} u_k{}^T Q_u u_k}^{\text{Actuator behavior limitation}} \right)$$

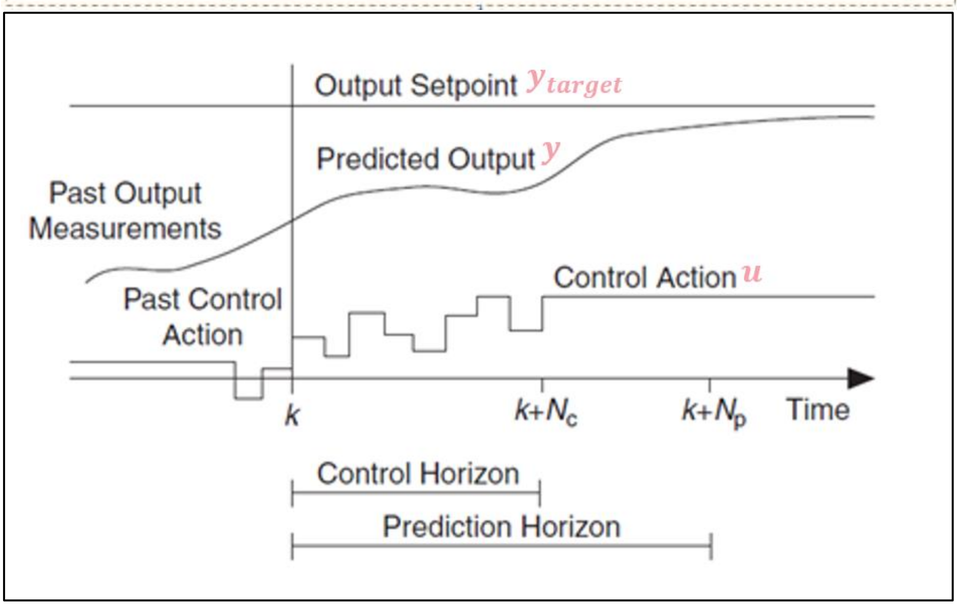


Figure 2 - MPC scheme

The above figure shows the working of an MPC and how it predicts the control action over time by solving the optimisation problem $\boldsymbol{u_{optimal}}$ as shown in the figure 2. At time K it calculates the best control action in the future for over a prediction horizon so that the output can reach the reference trajectory which is the setpoint or the required output of the system. The predicted output is the output of the system predicted into the future using the model used. The control action is based on minimizing a optimisation cost function $\boldsymbol{u_{optimal}}$ shown in the figure 2 to predict the control action in the future timesteps over a finite control horizon. This cycle is basically repeated over every timestep predicting new control action over the control horizon for new feed backs over time. This process can be explained by the following flow diagram:

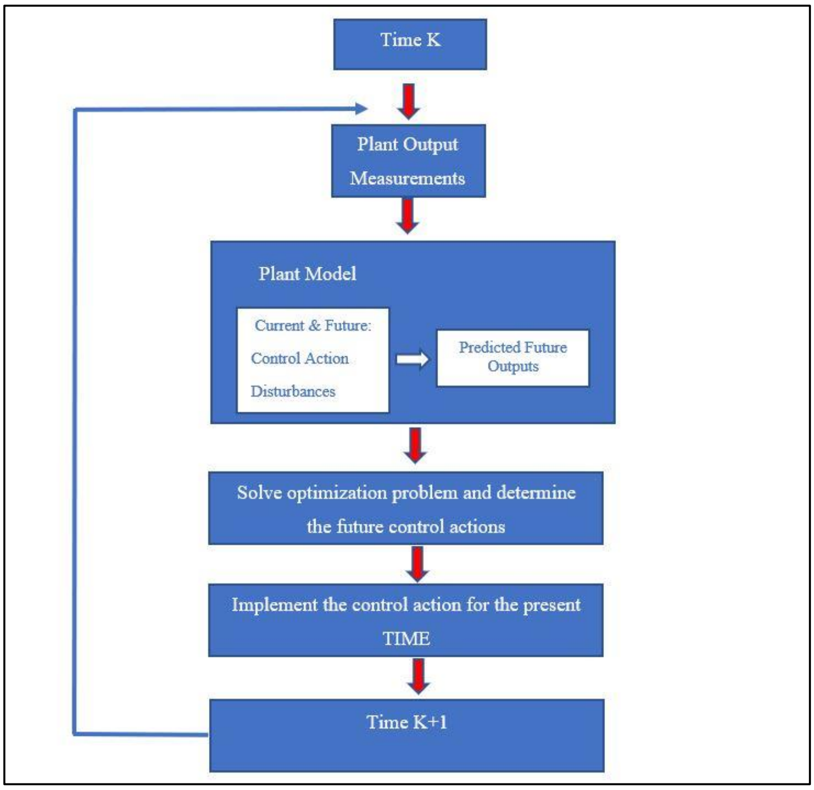


Figure 3-Process Flow of MPC[2]

The choice of using an MPC can be illustrated by the advantages it has over other controllers especially a PID(Proportional Integral derivative controller). PID is currently one of the most used controllers in the industry because of its ease of implementation. It doesn't need the dynamics of the system and calculates the output purely based on the error between the actual output and the setpoint. It is controlled using three gain parameters which characterize the behaviour of the controller . A schematic of the PID is shown below:

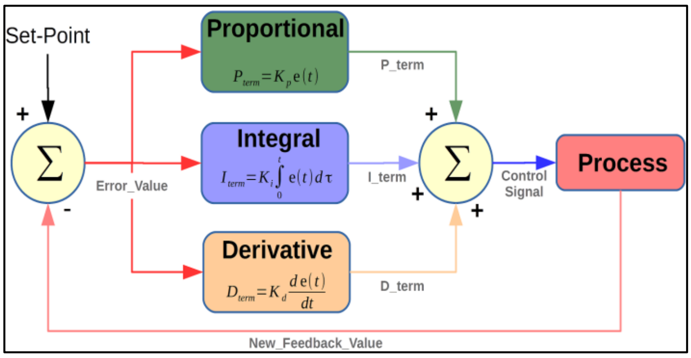


Figure 4-PID Controller

As it can be seen it has three gain parameters namely, Proportional gain K_p , Integral gain K_i and Differential gain K_d to control the response of the output[4]. These three parameters are tuned manually in most cases to achieve a smooth and fast transition of the output to the setpoint. But when the number of control actuators double as in our case where we have a throttle and a wastegate , the number of tuning parameters doubles up. So now it is required to tune two sets of tuning parameters simultaneously to target the setpoints which can be a very time consuming process and challenging. These limitations can be overcome by using a MPC controller as it simultaneously minimizes the cost function shown in figure 2 to find the actuator output at the next timestep.

In our project the MPC is implemented inside the Engine control unit(ECU) as a part of the logic to calculate the throttle and the wastegate. As the MPC uses a model of the system it needs measured inputs from the ECU for the model to predict the outputs. The MPC also uses the setpoint targets from the ECU and optimizes the control action of the throttle and wastegate

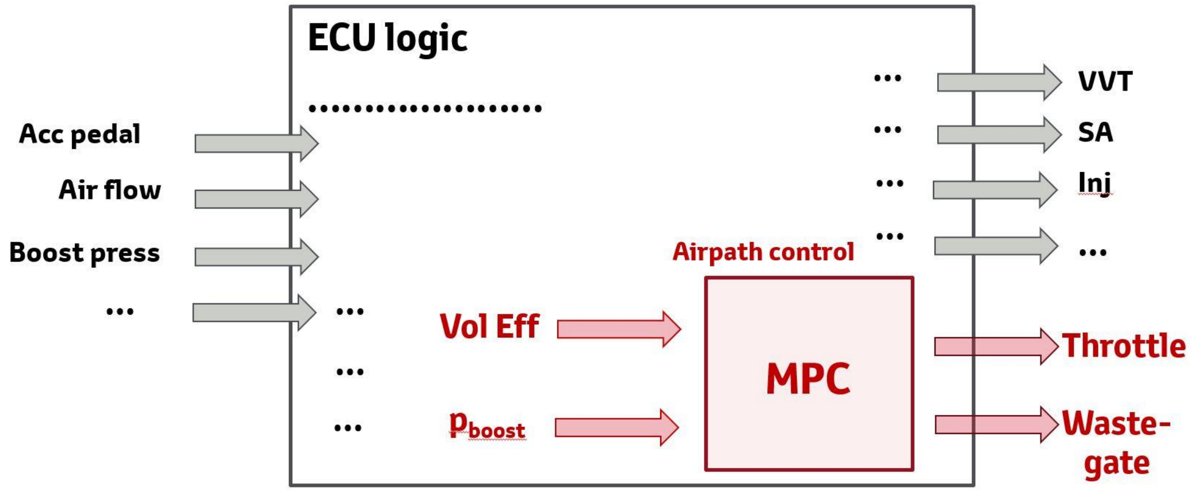


Figure 5 - MPC in Engine control logic

2.2. Development, Implementation and Testing of MPC:

It is now known that a MPC needs a model of the system to predict the future outputs. So in our case it needs a model which can predict the future outputs of the Engine airpath so the throttle and the wastegate can be controlled. Model creation is a part of the MPC development. A more detailed explanation can be supported by the understanding the workflow of MPC development.

The controller development follows a ‘V cycle concept’[5] workflow pattern which is shown in the figure below:

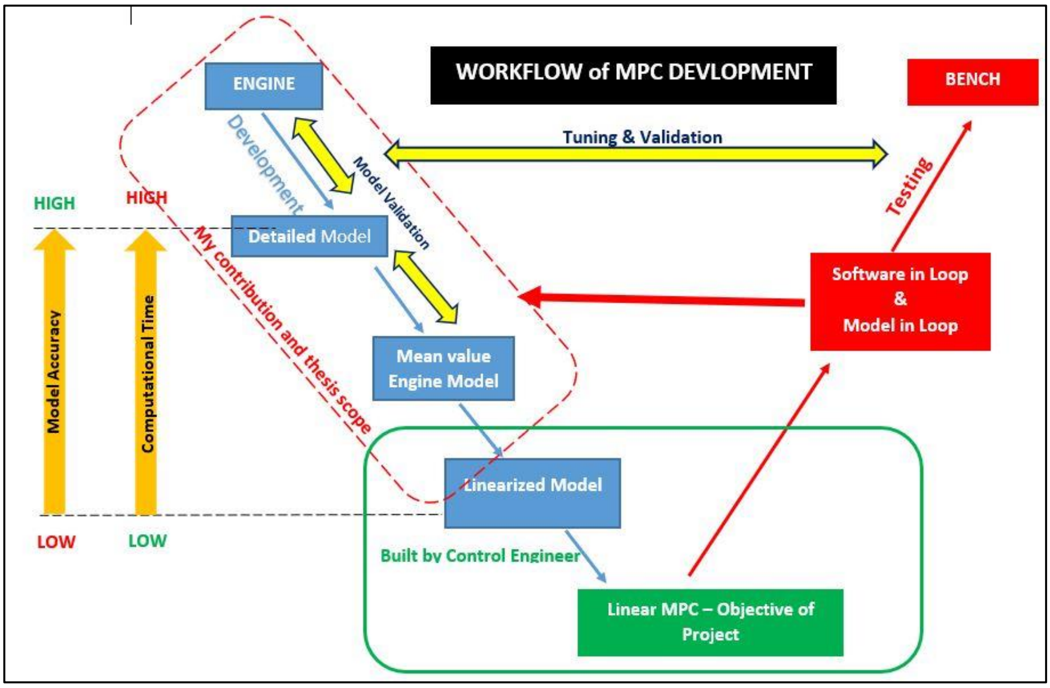


Figure 6-Project Flow

The workflow process can be separated into the Development phase which goes from the top down on the left side and the testing phase which goes from bottom up on the right side.

The development phase explains the model creation process for implementation of controller and can be explained by the following points:

- It starts with the engine on which the control needs to be implemented.
- The engine is modelled as a Detailed model which can reproduce the engine performance and overall 1-D flow behaviour(this doesn’t include detailed 3-D flow behaviour). These models need to be validated with the Engine to check for accuracy.
- The Engine model is simplified into a Mean value model and further into a linear model. The purpose of the simplification is :
    - The model should be linear to be used in controller.

  - The model should be fast because the controller needs to be quick enough to act on the engine actuators.

  The detailed model is very slow but simulates a very detailed output behaviour. As the models are simplified they lose accuracy but the computational time is less. We need to simplify them based on a trade-off between the computational time and the model accuracy. The computational time should be low enough but without losing a lot of accuracy.

- The linear MPC is developed using the simplified linear model which is the objective of the overall project but now it needs to be tested for performance.

The testing of the controller phase is explained below:

- The controller is tested in a toolchain test rig in MIL and SIL environments. The engine models (detailed model and Mean value models) are used as reference during the testing phase. The performance is evaluated and the controller is tuned based on the testing result.
- The performance of the controller is compared to the conventional control logic used. This helps us to validate and see if the controller is good or bad.
- After the controller is evaluated it is now ready to be implemented on the engine. The engine performance with the new controller is compared against the previous engine performance to record the controller performance.

From the workflow we can see that the Detailed and the Mean value models are used for MPC development and used in MIL and SIL testing of the controller. This calls for the validation of the models to ensure their accuracy . The detailed model is compared to the engine behaviour and the MVEM is compared to the detailed model behaviour. The validation process is the objective and scope of the thesis.

2.3. Validation Process Flow:

From the scope of the thesis it can be understood that the objective of the thesis is to validate the engine models. Before going any further it is important to understand the process flow of the validation.

The validation process flow is explained below with the schematic:

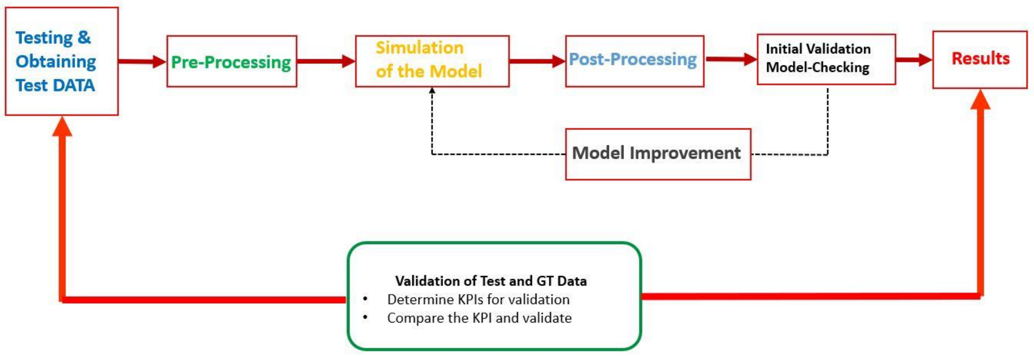


Figure 7-Validation Process flow

Testing and Obtaining Test data: Testing the engine is the first step as the model needs to be validated against the engine. The type of test procedures needs to be determined based on the objectives of the project and model validation purpose. Once the testing is done test data should be collected from the test bench for Simulation and Validation purposes.

Pre-Processing: This step involves converting the raw data from the bench to usable sources for simulating the engine models.

Simulation of Models: The models need to be simulated to obtain the outputs which characterize the model behavior.

Initial Validation-Model Checking : The first round of simulations should be test simulations as it should be made sure the models are working properly. A initial validation process is carried out to check the models. If the models have any error then model improvement needs to be done. Once the models are satisfactory from the initial validation the process can be used to obtain the test results.

Model Improvement: If the models perform abnormally the root cause of the problem should be analyzed and the models need to be improved. This is a loop which is repeated until the models produce satisfactory results.

Results: Once the models are satisfactory from the initial testing process now they can be simulated and postprocessed to obtain the Results of the simulation. These are model outputs which are compared to the outputs of the engine such as Boost Pressure, Intake Pressure, Volumetric efficiency etc.

Validation: This is final step where the model results are compared with the test data. Key performance Indexes(KPIs) need to be determined to effectively validate the model against the engine and each other.

3. Engine Models

As the objective of the project is to develop and validate the models before starting with the processes for validation, the background of the models is an important factor to know. The objective of engine models is to satisfy the requirements of the project which in this case it implementation of controller on airpath. A good model for this requirement should be able to simulate the flow dynamics of the system to predict the right pressures, mass-flow and volumetric efficiency. But it is to be noted that the complex 3-D flows and in-cylinder flow dynamics are not of interest to us as we are concerned more about the overall system flow behavior. For the purpose of this project 1-D engine models can be used as it can simulate the flow behavior along the engine airpath.

One dimensional(1-D) modelling is based on solving the engine conservation equations along the flow direction. This technique may be less accurate as the real engine has a more complex flow behavior and 3-D CFD flow models would be more accurate, but they account for very high computation time.

In a 1-D model the equations on conservation of Mass, Momentum and Energy are solved in one dimension along the major flow direction. The model consist of all the components where the flow occurs from the air-filter to the exhaust outlet. The following figure describes a 1-D engine model(Turbocharged) with its components.

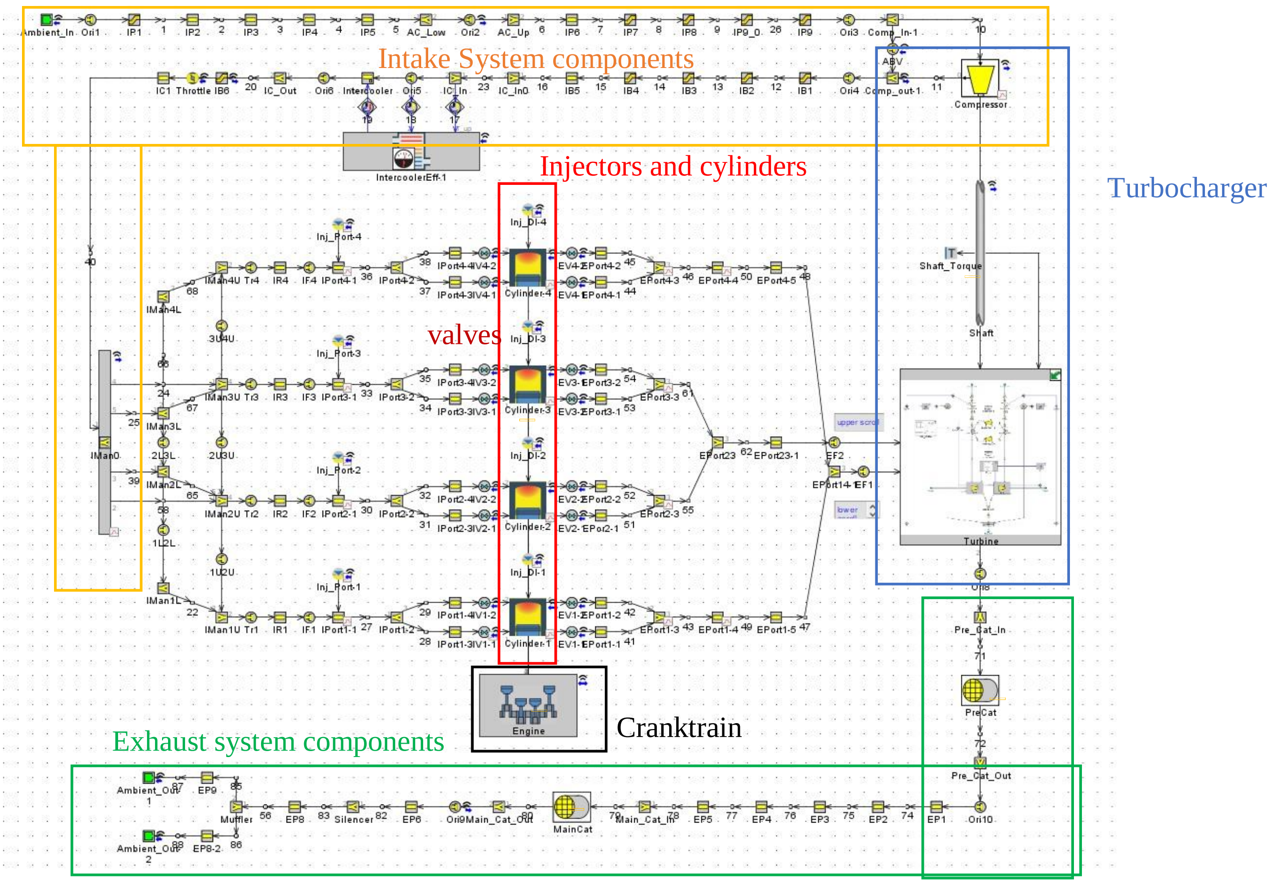


Figure 8 - Components of Engine Model

3.1. Modelling Theory of a 1-D Engine Model:

As it is a 1-D model all the quantities are averaged across the flow direction. The whole system is discretized into smaller volume and the discretization is done based on the needs of the model. Each pipe or flow component is divided into one or many volumes depending on the requirement and the accuracy of calculation. The volumes are connected by boundaries. The scalar variables such as pressure, temperature, density, enthalpy etc. are considered to be uniform over each volume. The vector variables such as mass flux, velocity, etc. are calculated for each boundary.[17]

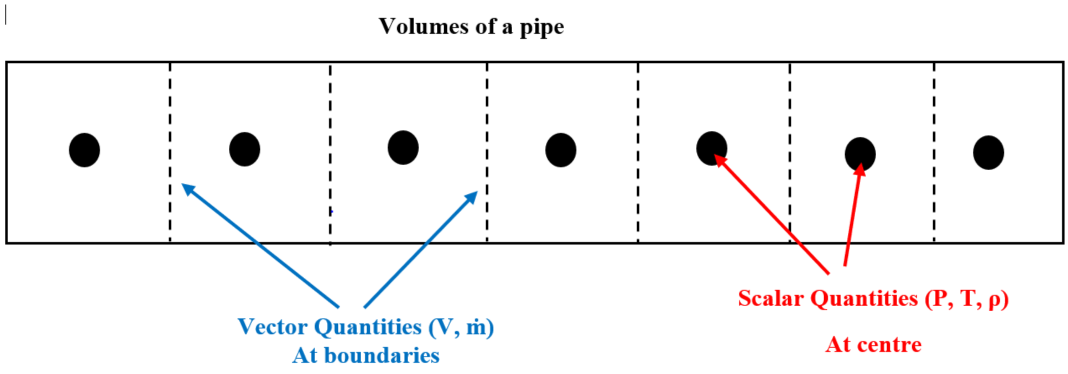


Figure 9-Flow volumes

3.1.1. Governing equations:

Conservation of mass:

This is given by the continuity equation which basically states that the amount of mass entering the boundary is equal to the amount of mass leaving the boundary considering the fluid is incompressible. For a compressible fluid the difference between the mass entering and leaving a system is the accumulation of mass inside the system.

Continuity: $\frac{dm}{dt} = \sum \dot{m}$	Equation 1

Conservation of Momentum:

The momentum conservation equation is based on the concept that the net resultant of all the forces acting on a boundary is equal to zero. There are two types of forces that act on the fluid element:

- Body forces - Forces that act on the whole mass of the element.
- Surface forces – Forces that act on the surface of the boundary such as pressure distribution on the surface, the shear forces on the surface such as fluid friction etc.

Momentum: $\frac{d\dot{m}}{dt} = \frac{dpA + \sum_{boundaries}(\dot{m}\,u) - \frac{4Cf\frac{(\rho u\lvert u\rvert)}{2}\frac{dxA}{D}} {} - Kp\,((\rho u\lvert u\rvert)/2)A}{dx}$	Equation 2

Conservation of Energy Equation:

The conservation of the energy is the rate of change of energy inside the volume. It is the sum of the

- Net Heat flux in and out of the volume
- Rate of work done on the volume element due to the forces such as the pressure force.

Energy: $\frac{d(me)}{dt} = -\frac{\rho dv}{dt} + \sum(\dot{m}\,H) - hA_s(T_{fluid} - T_{wall})$	Equation 3

Where:

$\dot{m}$ Boundary mass flux into volume

m Mass of the volume

V Volume

p Pressure

ρ Density

A Cross sectional flow area

A_s Heat transfer surface area

e Total specific internal energy

H Total specific enthalpy, H=e + $\frac{p}{\rho}$

h Heat transfer coefficient

T_{fluid} Fluid temperature

T_{wall} Wall temperature

u Velocity at the boundary

Cf Fanning Friction factor

Kp Pressure Loss

D Equivalent diameter

dx Length of mass element in flow direction

dp Pressure difference across dx

Pressure Calculation:

Calculation of the pressure can be confusing as the pressure can be calculated anywhere inside the volume. The solver normally calculates it at the center of the volume. Static pressure is the state variable that is calculated by the solver. Total pressure is calculated by post processing the results from pressure(static),velocity and fluid properties. Static pressure result has its disadvantages for example in the case of a venturi or in the case of an abrupt area increase where pressure recovery phenomenon takes place. In these cases the pressure rises when there is an increase in diameter due to pressure recovery. At these areas some of the kinetic energy due to lower velocity will be converted into pressure recovery and the rest will be expansion losses.

A potential mistake especially for gases is the calculation of total pressure. The definition of pressure for compressible flow should be used instead of the simple incompressible flow definition for pressure(Bernoulli equation).

The equation for the pressure calculation is as follows :[6]

$p_o = p + \frac{\rho u^2}{2}\ (1 + \frac{M^2}{4} + (2 - \gamma) - \frac{M^2}{24})$	Equation 4

Where

p_o Total stagnation pressure

p static pressure

ρ static density

u velocity

M Mach number

γ specific heat ratio

3.1.2. Solver Methods

Talking about the solver methods, GT-suite uses two solver methods namely:

- Explicit time integrator method
- Implicit time integrator method

The solution variables of the explicit method are massflow, density and internal energy. The solution variables of the implicit method are massflow, pressure and total enthalpy.

Explicit Method:

The solution variables in the explicit method are calculated using the conservation equations. The right hand side of the equations are calculated based on the values from the previous timestep and this allows the value at the new time to be calculated by the integration of that derivative over the new time step.

The following equation is an example of explicit method:

$Y(t + \Delta t) = F(Y(t))$	Equation 5

This method is well suited for unsteady flows where a high degree of resolution is needed to account the extremes of the flow behavior. It can make accurate predictions of the pressure pulsations and it is useful when the prediction of the pressure wave dynamics is important. The drawback is large timesteps cannot be considered as it affects the accuracy of the calculation.

Implicit method:

This method solves the values of all sub-volumes simultaneously for the present and the future. This is by iteratively solving a non-linear system of algebraic equations.

$G\big(Y(t), Y(t + \Delta t)\big) = 0 \quad solving\ for\ Y(t + \Delta t)$	Equation 6

This is useful for systems which has no importance on the wave dynamics and pressure pulsations. Large time steps can be solved with this method which can decrease the computation load.

3.1.3. Discretization:

Talking about the discretization of volumes, there are two different ways its done. One is by splitting up the system into different separate pipes which is called Coarse discretization. Other is be discretizing a pipe or a component into multiple smaller volumes namely fine discretization. Coarse discretization with larger lengths will result in a faster computational time but at expense of accuracy. Finer discretization will lead to better accuracy but only until a certain lower limit. After that the discretizing into smaller lengths will only increase the computational time with little or no increase in accuracy. Choosing the type of discretization depends on the type of component and the complexity of the flow through them. Example for:

- The intake pipes from the air filter to the compressor which are very simple cylindrical tubes coarse discretization is a good way to go.
- The pipes in the intake manifold with varying cross sections, bends and flow splits. Discretizing them into finer volumes would yield better results.

Further it is explained about the models used in the thesis namely detailed and mean value models(MVEM);

3.2. Detailed Engine Model

The detailed engine model is a highly detailed one dimensional model of the engine containing most of the components and they are all modelled with the governing equations reproducing a detailed system behavior. For example the cylinder model in a detailed model is modeled based on the

- In cylinder flow dynamics
- Turbulent Intensity
- Swirl and tumble
- Scavenging

Which are all crucial factors to characterize a detailed cylinder behavior. The combustion model is accurate enough but not as accurate as a 3-D CFD model.

The detailed model is modelled based on the following three models depending on their importance in each component. The models are listed below with the component dependencies:

- Flow model – Intake and Exhaust, cylinders, Valves , Throttle , Wastegate , Turbine ,compressor.
- Combustion model – Cylinders
- Heat transfer model or thermal model – Cylinders
- Mechanical model – Cylinder friction and Crank-train.

Flow models: Components with flow models are manifolds, pipes , Valves , Ports etc where the flow behavior is the important characteristic. The valves and ports use discharge coefficients to calculate the effective flow area. The discharge coefficients are predetermined from experimental data or 3D CFD simulation. Another important part of the flow model is the turbocharger as we are using a turbocharged engine for testing. The engine uses a twin scroll turbocharger . The turbocharger model is designed with two scrolls where the flow is characterized by using maps. The exhaust flow from cylinders 1,4 and 2,3 are split into the lower and upper scroll respectively. There is a map to characterize the cross flow between the scrolls to determine the flow of air over the turbine. The turbine is connected by a shaft component to the compressor which is in the intake part of the model. The compressor has a map to determine the pressure ratio across it creating boost.

Combustion Model:

The GT-model uses a two zone combustion model where the cylinder space with the charge air mixture is divided into a burned and unburned zone.

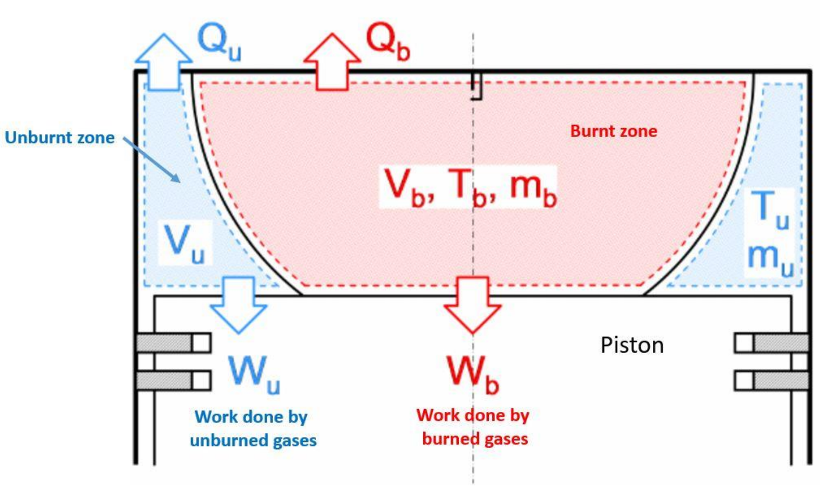


Figure 10 - Two zone Combustion model

Unburnt zone: It is total unburnt fuel air mixture inside the cylinder. At the beginning of combustion all the mass inside the cylinder is considered unburnt including residual gases from previous cycle.

The conservation of energy equation can be applied to the burnt zone by the following equation:

$\frac{d(m_u e_u)}{dt} = \underbrace{-p\frac{dV_u}{dt}}_{\text{Pressure Work}} \underbrace{- Q_u}_{\text{Heat release rate}} + \underbrace{\left(\frac{dm_f}{dt} h_f + \frac{dm_a}{dt} h_a\right)}_{\text{Combustion}} + \underbrace{\frac{dm_{f,i}}{dt} h_{f,i}}_{\text{Addition of enthalpy from injected fuel}}$	Equation 7

Burnt Zone: As the time progresses mass transfer occurs between the unburnt zone to the burnt zone. And the energy of the unburnt and burnt zone is calculated using the conservation of energy equations for both the zones separately. The burn rate is calculated using a SI-Turbulent flame model which is calibrated based on the [10] [11] [12]

- Laminar Flame speed
- Entrainment and burnup

Once the mass from unburned zone is transferred into the burned zone the chemical equilibrium calculations are done considering the burned zone as one lumped volume. The equilibrium concentration of the species strictly depend on the current burned zone temperature. Once the new composition of the burned zone is obtained the internal energy of all the species is summed up to give the total energy of the burned zone.

$\frac{d(m_b e_b)}{dt} = -p\frac{dV_b}{dt} - Q_b + \left(\frac{dm_f}{dt}\ h_f + \ \frac{dm_a}{dt}\ h_a\right)$	Equation 8

The subscripts u and b denotes if the quantities belong to the burned or unburned zone.

m - zone mass

e - zone energy

p - cylinder Pressure

V -Zone Volume

Q - Zone heat transfer rate

m_a - Air mass

m_f - Fuel mass

h_f - Enthalpy of fuel mass

h_a - Enthalpy of air mass

$m_{f,i}$- Injected fuel mass

$h_{f,i}$ -Enthalpy of injected fuel mass

Thermal models: The heat transfer models are important when it comes to designing the cylinder model where the heat from the piston is transferred to the cylinder walls as loss of energy. This can affect the in-cylinder pressure leading to an inaccurate cylinder model. The heat transfer inside the cylinder is based on the in cylinder flow model. The cylinder is broken up into multiple regions where the radial velocity, axial velocity and swirl velocity are calculated based on cylinder geometry and flow rate through valves. The in-cylinder heat transfer coefficients are based on the Woschni model[9]. The basic concept of the Woschni model states that, the average gas velocity should be proportional to mean piston speed.

$h_c = \frac{K_1 p^{0.8} w^{0.8}}{B^{0.2} T^{K_2}}$	Equation 9

h_c = Convective heat transfer coefficient(W/m^2K)

B = Cylinder Bore(m)

K_1 & K_2 = Constants

P = Cylinder pressure(kPa)

T = Cylinder Temperature(K)

W=Average cylinder gas velocity as given below

Table 1- K_1 & K_2 values for different Woschini models[8]

Woschini model	K_1	K_2
Woschini GT	3.10426	0.50
Woschini classic	3.26	0.53
Woschini swirl	3.26	0.53
Woschini Huber	3.26	0.53

Woschini GT and Woschni Classic are independent of swirl in the Engine. Woschini swirl and Woschni huber are dependent on the swirl numbers at the center of the cylinder.

For the wall temperature calculations a finite element method is used when the user defines the cylinder geometry , coolant and oil temperatures. The values can also be imposed for the wall temperature assuming typical values at full load.

Mechanical models: This includes the power transfer through the solid moving components such as the piston , connecting rods and the crankshaft. From performance point of view the Engine friction is an important attribute which should be modelled. The mechanical friction is modelled using the Chen-Flynn model using the following equation:[14]

$$FMEP = C + (PF * P_{max}) + \left(MPSF * Speeed_{mp}\right) + (MPSSF * {Speeed_{mp}}^2) \quad \text{Equation 10}$$

Where

FMEP - Frictional mean effective pressure

P_{max} - Maximum cylinder pressure

$Speeed_{mp}$ - Mean piston speed

C - Constant of FMEP

PF - Peak cylinder Pressure factor

MPSF - Mean Piston Speed factor

MPSSF - Mean Piston Speed square factor

This model states that the total engine friction is the function of peak cylinder pressure ,mean piston speed and mean piston speed squared.

The figure below shows the type of models used in the components of the 1-D detailed model used in this project.

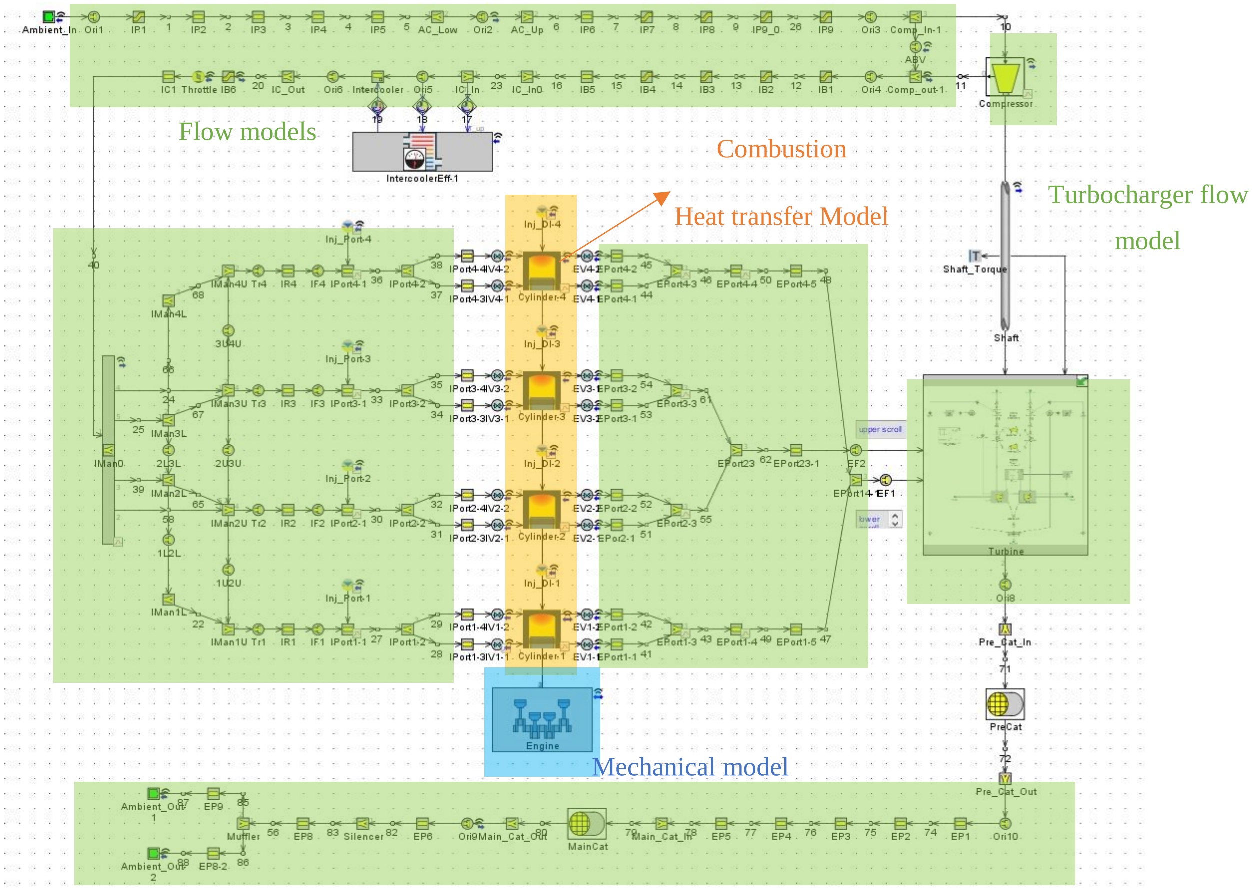


Figure 11 - Detailed Engine Model

The detailed model can capture the accurate wave dynamics and pressure pulse behavior of the system. This is because the volumes are discretized and the time steps are usually a fraction of the crank angle . So the pressure can be seen pulsating across every cycle of the engine. The cycle to cycle variations are also observed accurately in a detailed model.

Though the detailed model has high accuracy can simulate the wave dynamics of the flow in the engine it has its downsides. It has high computational time and computational load. Since the time step is very small and the amount of governing equations are high the detailed model takes a lot of time to solve and produce results. This can be a huge disadvantage when the wave dynamics and cycle behavior of the system are relatively unimportant. Example, while executing really long transients and during testing of control systems behavior. Also the wave behavior of the detailed model cannot be linearized which is a requirement to build the linear model for control purposes. So a MVEM needs to be developed.

3.3. Mean value Model:

This kind of models are used where less computational time is of high importance and detailed characterization of engine processes are relatively less important. The major characteristics of Mean value models are as follows:

- The mean value model calculates the averaged values of the parameters over the engine cycle. It implies that the values of pressures , mass flow etc. are constant over a cycle.
- Use of Maps to characterize the cylinder air flow and distribution of fuel energy which means the cylinder model is not modelled as in a detailed model using governing equations .This results in really fast execution of the cylinder model because the breathing and combustion processes are not predicted and values are obtained from maps. All the models used in the detailed model section for the in-cylinder flow, cylinder heat transfer are of no concern in the Mean value models.
- The volumes in a mean value models are lumped together. This results in lesser components and lesser boundaries. Since we are using maps the flow is steadier than the detailed model, lumping the volumes will not affect the flow simulation to a great extent.

In a mean value model there are three quantities that are used to characterize the engine cylinder performance. They are:

- Volumetric efficiency of the cylinder which is used to determine the mass flowrate of air into the cylinder.
- The indicated efficiency which describes the amount of fuel energy which will be converted into work on the piston.
- Exhaust temperature which determines the amount of fuel energy that will be transferred into the exhaust gases.

These quantities are defined as maps. Or they can be calculated by an external control system and actuated. This external method is used when the quantities are a function of two or more than two different input variables.

This means that the quantities are not calculated based on the actual governing equations but instead calculated based on the best correlation on the dependent variables. The FMEP is also calculated based on this method. In a detailed engine model the friction is modelled based on the peak cylinder pressure as mentioned before. Since the mean value model does not model the breathing or combustion and is essentially a steady flow component the peak pressure in a cycle is much less than the actual detailed cylinder. Thus the friction cannot be dependent on the peak pressure of the cylinder as it can lead to inaccurate results. The FMEP can be modelled as a map , dependent on two or more variables to make it more accurate.

Because the detailed wave dynamics are of no concern in mean value models it is possible to highly reduce the computation time by increasing the size of the time step. The time step of a normal detailed model is limited to a maximum of one crank angle degree. In a mean value model it can be increased up to 15 crank angle degrees. This can reduce the computational load and time greatly.

From the objective of the project we know that the MPC uses a linear model. The linear model cannot be created from the Detailed model as the wave behavior of the detail model cannot be linearized. It's necessary to create a mean value model which is linearized using a Linearization tool in GT-suite.

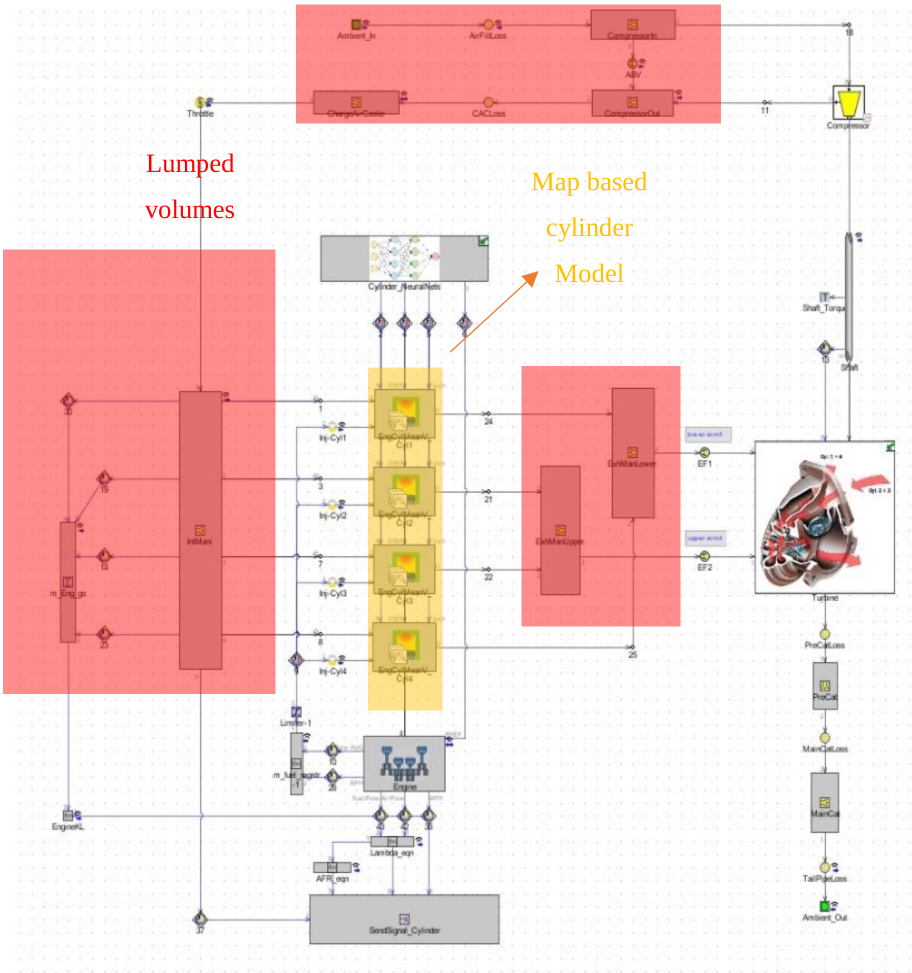


Figure 12 - Mean Value Engine Model

4. Testing of Engine:

The Testing of the engine is the first step in the validation process as the model performances should be validated against the engine performance. The type of the tests needed to be performed is one of the major requirements for the validation process. Determining the type of tests require understanding the objectives of the Project which can be recalled as follows:

- Construction of Engine models for the development of MPC – To predict the future system behavior.
- Use of the models in MIL/SIL environment for the testing of MPC – As reference engine.

This makes it clear that the models should be accurate enough to satisfy the above objectives and we should choose the test specifications so that the tests can ensure :

- ➢ Ability to validate the model and the engine at steady conditions. The model has to predict the flow related parameters such as right Mass flow, Volumetric efficiency and Boost pressure over the engine operation zone.
- ➢ Ability to observe the right transient behavior of the model and the engine. The transient behavior of the models are important as the controller works on the actuator over time in real time scenario. Only if the models are accurate enough to predict the transient engine flow behavior they are suitable for the purpose of the project.

The steady and transient conditions can be briefly explained by the following figure:

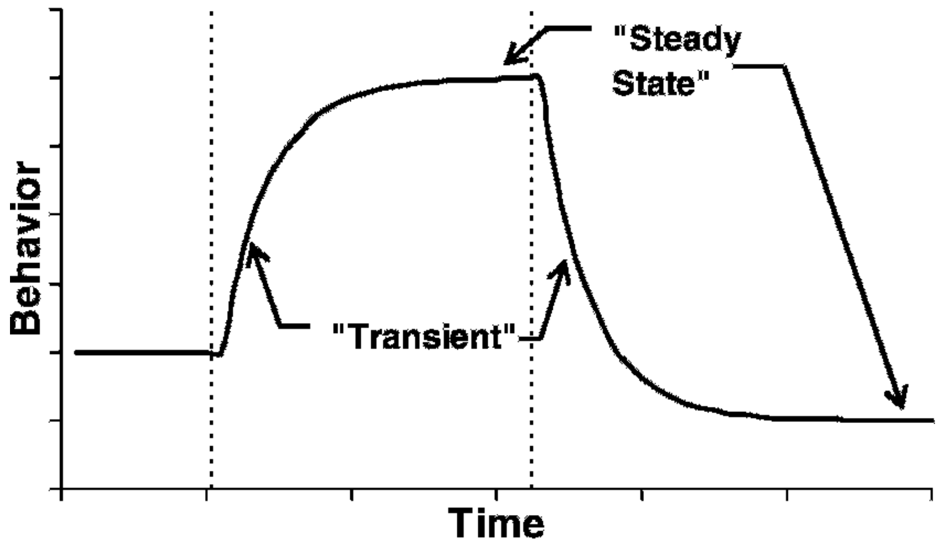


Figure 13 - Steady state vs Transient conditions

The engine used for the controller implementation is a 2.0 liter 4 cylinder twin scroll turbocharged - SI engine.

The engine is tested on the Testbench where it is coupled to the dynamometer. The schematic of test setup is shown below:

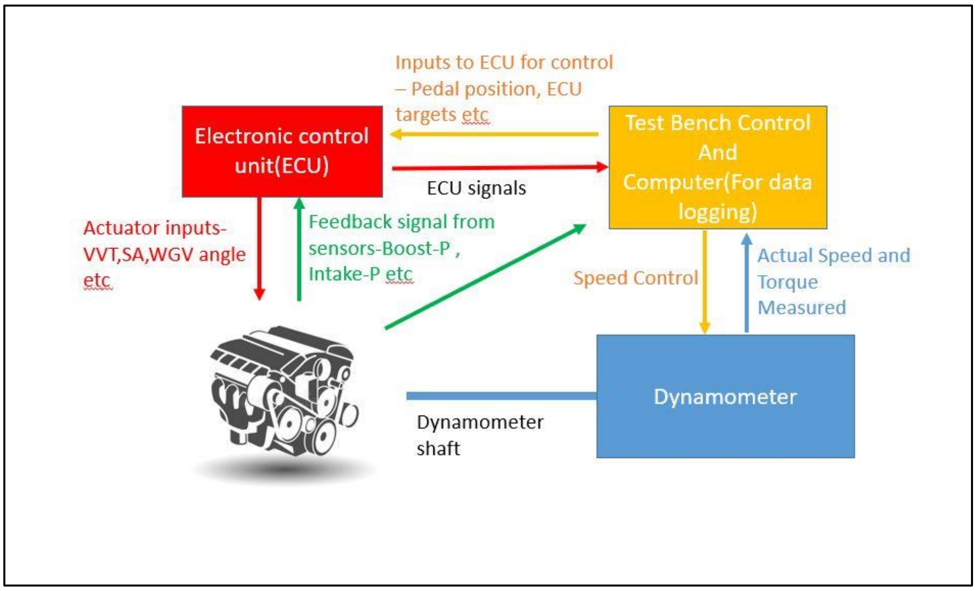


Figure 14 - Test Bench setup and control

The dynamometer loads the engine controls the speed at which the engine operates. The engine characteristics are measured for different loads at various speeds. This type of testing is called “Speed control testing” as the Speed is controlled by the dynamometer. Basically it implies that the engine speed can be set by the testing person and the engine performance parameters can be measured for different loads controlling the speed.

For the measurement purposes sensors are mounted on the engine for measurement of Engine speed, Boost pressure, Mass-flow , Intake manifold pressure , Compressor speed etc. to obtain data for analyzing engine behavior. The engine comprises of a ECU(Electronic control until) which calculates the required throttle angle, Wastegate angle , Intake VVT angle, Exhaust VVT angle, Spark advance angle, Injection Quantity, Injection timing etc. The ECU is fed with various maps and logics to perform these operations to ensure optimum engine performance. Throttle and Wastegate are based on feedback control using a PID controller . The ECU obtains feedback from the sensors for mass flow and Boost pressure and uses the controller to target the setpoint for the Boost pressure and Mass-flow to obtain the required load.

Note the testing setup depends on the requirements of the project. As we are implementing controller on the airpath we need the flow based data such as the mass flow and pressure measurements in the intake and exhaust. The position of the actuators are needed too. The exhaust system components such as three way catalyst, Pre-cat etc. can be removed as they serve no purpose for our project. The back pressure can be adjusted using a flap type valve in the exhaust line to match the right back pressure with the exhaust systems attached.

4.1. Testing at steady

Steady testing of the engine refers to taking the Engine to a certain speed and load and let the engine run at that point until it reaches steady state. Steady state is reached when the output parameters of the engine remains stable and don't change anymore as seen in figure 13. This testing is done over the operating range of the engine which is the Speed-Load curve. The data from the steady testing is used for model building especially throttle and waste gate model where the Coefficient of discharge is modelled based on the valve angle and pressure ratio.

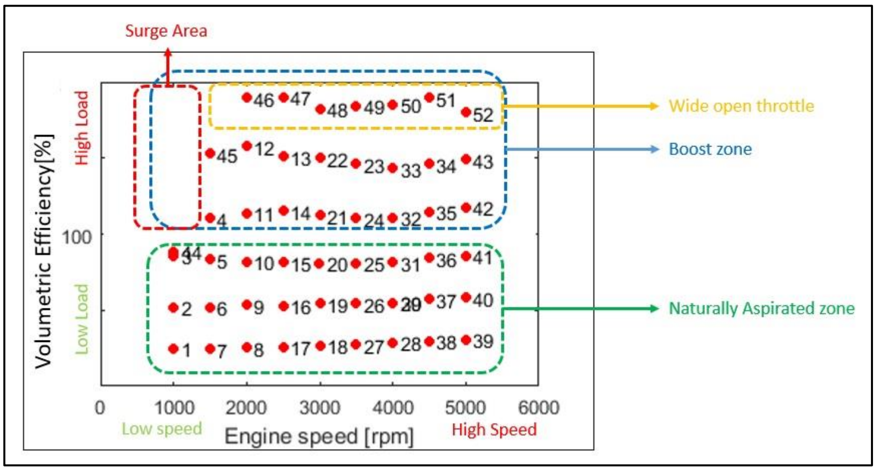


Figure 15- Operating Points

The operating zone can be used to characterize under what conditions the engine is operating. The operating range can be divided into zones based on Volumetric efficiency and Speed.

- **Naturally Aspirated zone** : In this area the engine works without the effect of the turbocharger. There is not enough energy in the engine exhaust to spool the turbine to generate boost and the engine breathes using the atmospheric pressure. The breathing efficiency is low in this area as the flow is obstructed by the Compressor and Turbine.
- **Boost Zone** : This is the area where the turbocharger works and enough Boost is generated to increase the pressure in the intake above the atmospheric pressure. In this zone the volumetric efficiency is higher than 100% as the density of compressed air is greater than density of air at atmospheric pressure.
- **Wide open throttle points** : These are the points on the maximum load curve. The engine produces its maximum torque and volumetric efficiency at these points.
- **Surge Area**: This is place of low speed and high load of the engine where the mass-flow is too low that it causes low frequency pulsating flow causing compressor instability. This can cause severe mechanical vibrations in the system. So the engine should not be operated in this area.
- **Low Speed** : This is the minimum speed points at which the Engine is tested. Operating on speeds lower than this can cause slower combustion and pre ignition leading to knock in the engine. So it's avoided.
- **High Speed** : This is the max speed points at which the Engine is tested. At low load the mass-flow is high but the pressure is very low and the density of the flow is low. At high load and high speed the flow characteristics change as the mass flow is very high and the pressures are high.

By testing at these points we can ensure the model is capable of predicting the steady engine characteristics under the different operating zones over the operating range.

52 points were chosen under the speed load curve and the engine was tested at these points.

Procedure of Testing the Engine at steady:

- Set the Engine bench to speed control(Impose the engine speed). Target the required speed
- Target the load with the known Accelerator pedal position value.
- Wait until the Pressures, Mass-flow and Temperatures reach a steady value.
- This procedure is repeated for all the 52 points and the DATA is logged. The steady state convergence takes normally around 10-20s .

The results of the Steady testing is used to create the mapping of the Engine parameters. Most importantly the values are useful to create Engine models. Especially the throttle and wastegate model. The flow and pressure values are used as reference for model building.

4.2. Testing at Transient

Transient Testing refers to testing the engine for different responses over the time where the engine behavior changes continuously as shown in figure 13. The Engine is tested taken from one point to another under the operating range depending on the requirements of the test. Transient tests helps to understand the Engine and model behavior in more detail as they help to understand the response of the Engine and model at close to real time engine operation. The right kind of transient test cycle or response is needed to validate the model based on our needs. For example drive-cycles are used for emission tests as they cover the behavior of the engine in most real time transient driving conditions. Other kind of transient tests are based on the requirements of the measurement being done such as knock , vibration analysis etc.

In this case we are testing the engine for obtaining the flow behavior for model validation. The test needs to make sure the models are capable of predicting the right transient engine airpath behavior for the actuator response. One of the test methods which can help us study this closely is a STEP response test. Performing an instantaneous step test on the engine from one operating point to another can help us compare the engine response and eventually compare the model response to it.

The advantages of performing a STEP test are :

- It provides good standard KPIs for analyzing the transient behavior.
- It’s a simple test to perform.
- We can concentrate on the zones and points where we need to perform the STEP. This helps one to analyze the engine better at the area in the operating range where analyzing the engine behavior is most required.

4.2.1. STEP Test specification

The advantages and requirements of performing a STEP test were mentioned in the previous section but the kind of STEP that needs to be performed is crucial for our validation requirements. Before concluding the type of STEP that needs to be performed it's essential to take a view of how the engine control works.

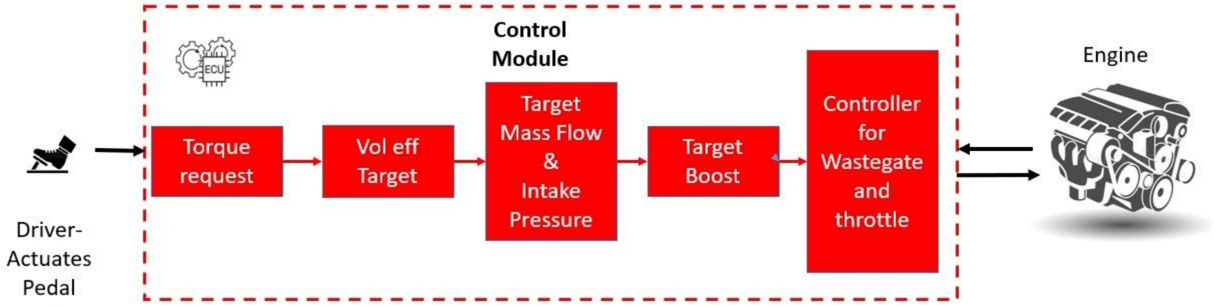


Figure 16 - Engine control Flow

Looking at the above figure we can see that the logic works based on reaching a volumetric efficiency target which is determined by the pedal position of the driver. Steps can be performed either through the actuators such as Accelerator pedal, Throttle, Wastegate etc. or through modifying the target values of Volumetric efficiency , Boost Pressure etc. inside the control module.

To determine the type of STEP to perform we need to analyze the engine behavior when the load on the Engine changes. So, load steps were performed on the Engine to record the different actuator behavior when the engine moves from one operating point to another.

Three variants of load STEP tests based on input targets were chosen for the testing:

- Accelerator Pedal Step : To check the engine transient behavior when ECU works with standard inputs and logic by controlling the STEP input through the accelerator Pedal position.
- Vol Efficiency Step with Standard and Constant Boost Target : Checking the engine behavior when a step is performed by modifying the target volumetric efficiency.
- Boost Step with Standard and Constant Volumetric efficiency target: The Boost target is modified to perform a load step.

The next specification part is choosing the zones where these STEPs need to be performed. To choose a more reliable testing zone under the operating range it was decided that the STEPs can be performed in the zones where the WLTC,TUV,NEDC6,RPA95,US06 drivecycles operate. The drivecycle points were plotted over the operating range to choose the speeds and loads for the testing.

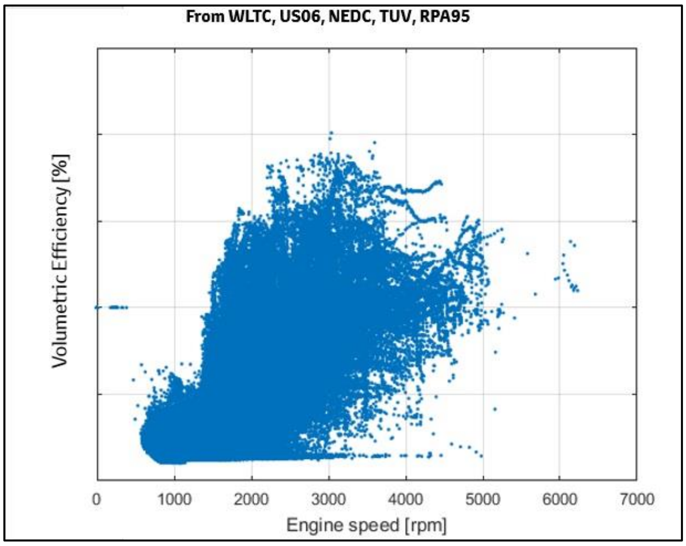


Figure 17 - Discretized points from drivecycle over operating zone

Since it's a turbocharged engine it's important to validate the model behavior when it transitions from the naturally aspirated to the Boost area and the behavior when it goes from Boost to Boost points.

To note: Steps were performed with nominal spark advance and 5 degrees retarded spark advance. Since during the HIL testing the engine is operated under 5 deg spark retard for safety purposes the same was included in the test specifications.

4.2.2. Accelerator Pedal STEP

The STEP is performed by controlling the accelerator pedal position through the bench controls. Doing a step on the accelerator does not affect the standard logic of the ECU. The ECU uses the standard logic to calculate the required volumetric efficiency , Boost pressure , intake pressure , Massflow ,Throttle angles , Spark advance ,Valve timing , Injection rates etc. This resembles a system where how the engine works in a real environment where no ECU bypass is there.

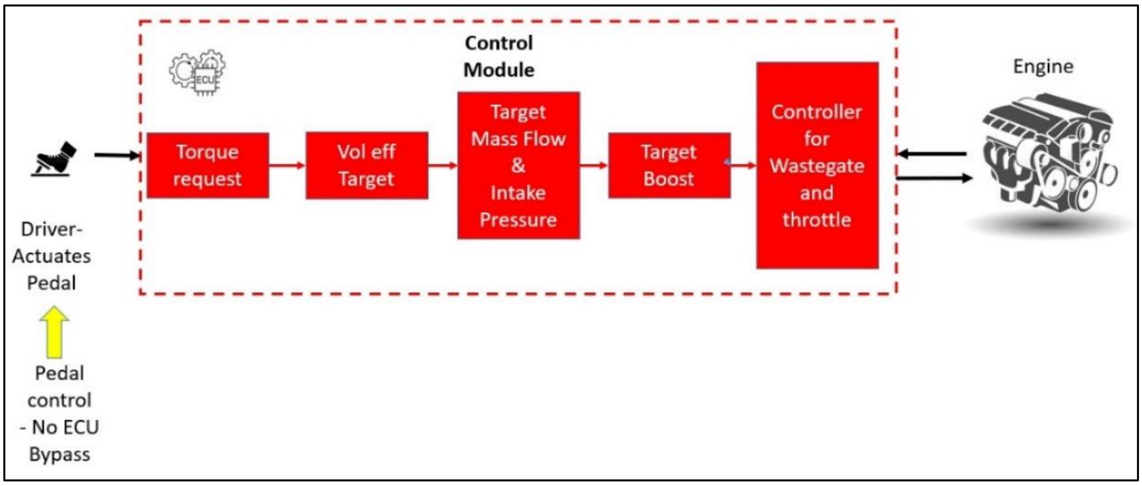


Figure 18 - Accelerator pedal STEP Control

Test Procedure:

- Set Bench Dynamometer to run the engine at constant speed.
- The accelerator pedal position is found out to match the initial and final load/Vol eff between which the step is performed.
- The STEP is performed by changing the pedal position value from the Bench until the engine stabilizes at a steady value at the final point and a Step down is performed in the same manner.

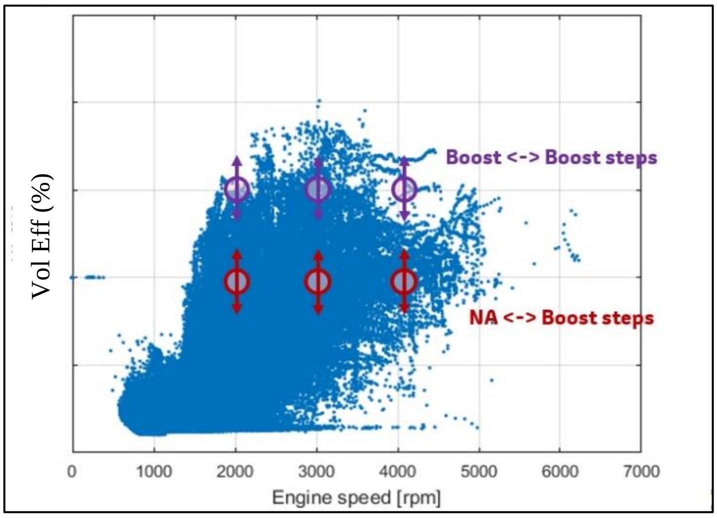


Figure 19 - Accelerator Pedal Steps

Table 2 - Accelerator pedal step-Test matrix

STEP TYPE	Engine Speed	Spark Advance	NA-Boost (70-120% Vol Eff)	Boost-Boost (130-170% Vol Eff)	Total
Accelerator Pedal step	2000	Base		1	
		"5 deg retarded"	1	1	
	3000	Base		1	
		"5 deg retarded"	1	1	
	4000	Base		1	
		"5 deg retarded"	1	1	
		Total	3	6	9

4.2.3. Volumetric Efficiency STEP:

The Load step is performed by changing the volumetric efficiency target values bypassing the Control module. The step is performed with constant and standard initial Boost target. By providing a constant initial boost target we force the Module to match the boost when the step is done on the volumetric efficiency target. This helps us to test the engine behavior at a wider range of Actuator responses.

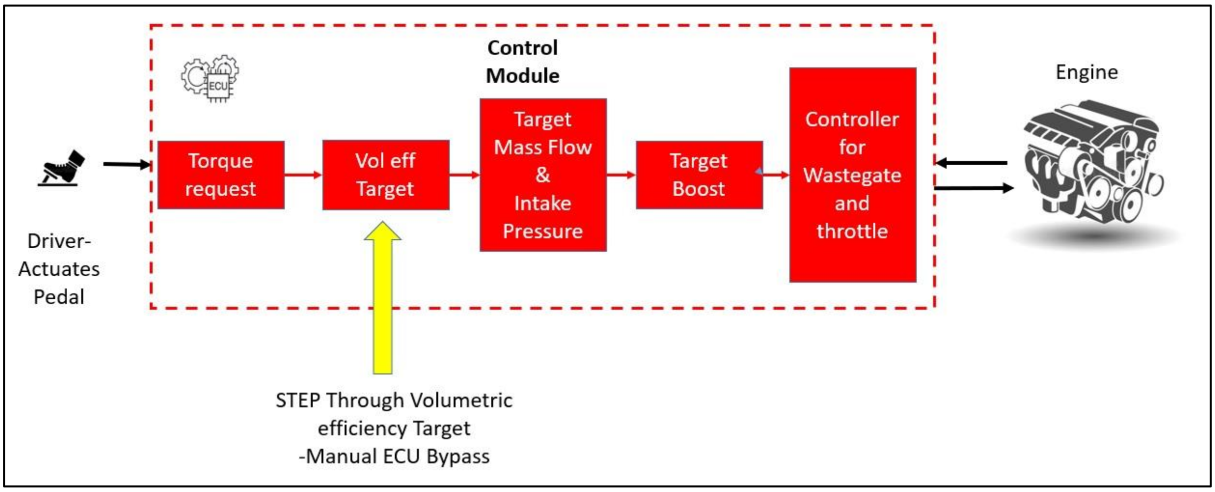


Figure 20- Volumetric efficiency STEP Control

Procedure:

- Set Bench Dynamometer to run the engine at constant speed.
- Determine the initial and final Volumetric efficiency values.
- Run the engine with the initial vol eff target until it reaches steady state .
- Do a step by changing the vol Eff target through the ECU and wait until the Engine reaches steady state.
- Do the same procedure with bypassing the Boost target by a constant boost value initially and making a step.

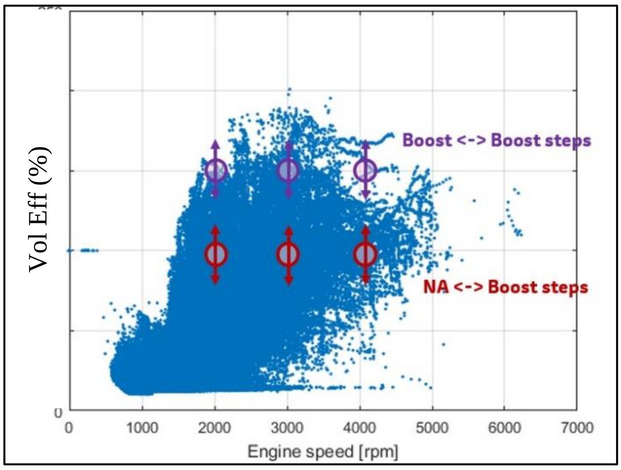


Figure 21 - Volumetric Efficiency steps

Table 3-Volumetric Efficiency Step Test Matrix

STEP TYPE	Engine Speed(rpm)	Initial Boost request	Spark Advance	NA-Boost (70-120% vol Eff)	Boost-Boost (130-170% vol Eff)	Total
Volumetric efficiency step	2000	Standard	Base	1	1	
			"5 deg retarded"	1	1	
		Constant	Base	2	1	
			"5 deg retarded"	1	1	
	3000	Standard	Base	1	1	
			"5 deg retarded"	2	1	
		Constant	Base	1	1	
			"5 deg retarded"	1	1	
	4000	Standard	Base	1	1	
			"5 deg retarded"	2	1	
		Constant	Base	1	1	
			"5 deg retarded"	1	1	
				15	12	27

4.2.4. Boost step:

The step is performed by modifying the Boost pressure target in control module. It requires bypassing the control module. The Boost pressure target steps are performed with constant vol eff and standard initial volumetric efficiency targets. By doing this we provide the engine with a higher boost target increasing the load but try to force it match the initial volumetric efficiency. This helps us gather information on the engine behavior for a wider range of actuator behavior. This step is performed only in the Boost area of the Engine.

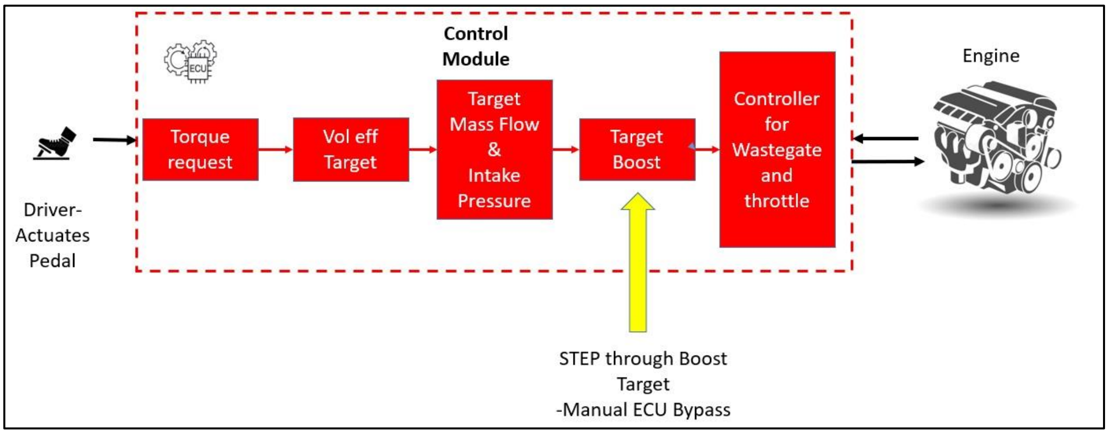


Figure 22 - Boost STEP Control

Procedure:

- Set Bench Dynamometer to run the engine at constant speed.
- Find the Boost Pressure values to match initial and final load/Vol eff.
- Run the engine to match initial Boost target and initial load until it reaches steady state.
- Do a step by changing the Boost target through the ECU and wait until it reaches steady state and do the vice versa.
- Follow the same procedure by setting the initial vol-eff target to be constant and perform a step on Boost pressure target.

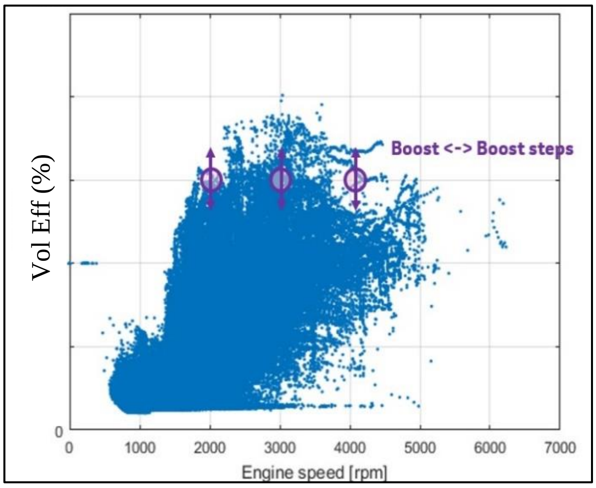


Figure 23 - Boost Steps

Table 4 - Boost Step Test Matrix

STEP TYPE	Engine Speed(rpm)	Initial Vol-Eff request	Spark Advance	Boost-Boost (130-170% vol eff)
Boost step	2000	Standard	Base	1
			"5 deg retarded"	1
		Constant	Base	1
			"5 deg retarded"	1
	3000	Standard	Base	1
			"5 deg retarded"	1
		Constant	Base	1
			"5 deg retarded"	1
	4000	Standard	Base	1
			"5 deg retarded"	1
		Constant	Base	1
			"5 deg retarded"	1
			Total	12

5. Pre-Processing of DATA:

Pre-Processing refers to the conversion of the raw data obtained from the tests into usable data to be used for simulation and analysis purposes. Test Data from the Engine Test bench are usually from two sources during a Test:

- From ECU: ECU records the data such as Variable valve timing, Injection Quantity, Spark advance, Injection timing , Throttle angle etc. It also records the massflow and pressures of some components based on the inbuilt sensors in the production engine. The Volumetric efficiency of the engine is calculated by the ECU based on the massflow and Engine speed. The targets such as Boost pressure target, massflow target etc. are obtained from the ECU.
- From Bench: The test setup is done by incorporating extra sensors on the engine for the measurement purposes of the test. A position sensor on the wastegate , Pressure and temperature sensors on the turbocharger scrolls are mounted. A speed sensor is mounted on the turbocharger. This data is obtained directly by the bench computer.

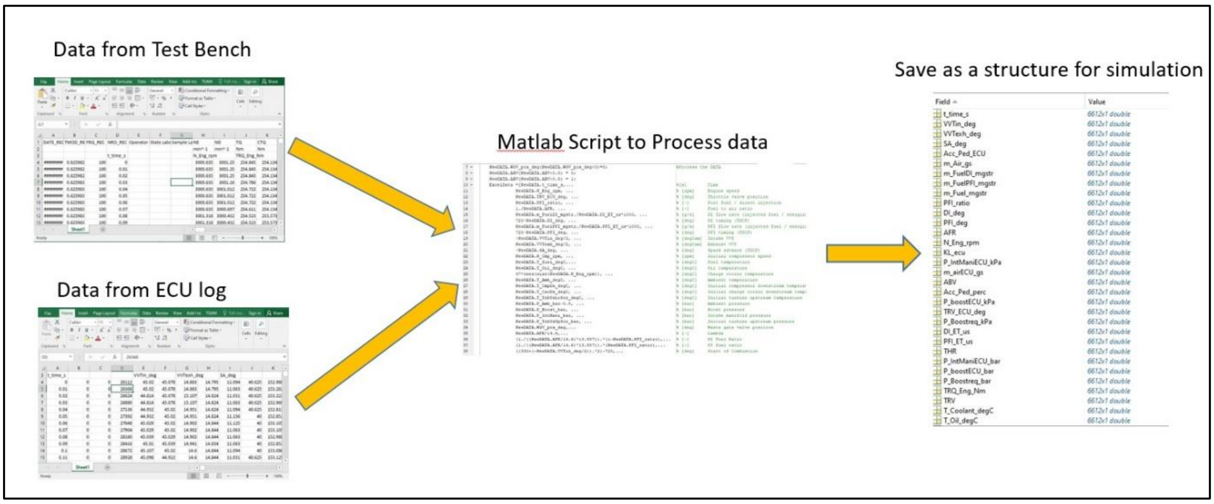


Figure 24 - Pre-Processing data for simulation

Preprocessing refers to transformation all the above data so they can be used for simulation purposes. This includes synchronization of data if needed and unit conversions of data values. The following data should be preprocessed:

- The Pressure data from sensors are logged in kilopascals and absolute pressure. The data from the ECU is logged in relative pressure. Uniformizing the units of the data are very important so it's easier to use them for simulation purposes.
- Data conversions should be done. This is especially for the position value of the wastegate. It is obtained as a voltage reading from the sensor and should be converted into angle value using the right conversion criteria. The relation between the wastegate angle-Voltage is derived from the sensor specification sheet.

- The injection rate should be calculated from the injection amount and energizing time from the raw data. The ECU records the injected injection amount in mg/stroke and the energizing time in microseconds. The injection rate is calculated by:

$$Injection\,rate = \frac{Injection\ amount\ (\frac{mg}{str})}{Energizing\ time(\mu s) * 1000}\ (\frac{g}{s})$$

- The Valve Timing ,Spark advance and Injection time should be angle shifted relative to the cycle duration.
- The lambda for the model should be calculated based on any corrections in AFR as the model stochiometric is assumed to be different from the Engine.

6. Simulation Setup in GT-Suite

To understand the simulation of the models it's essential to recall the testing of the engine from the testing chapter. It was explained that the control of engine goes from the user to the engine through the ECU. The Engine model used for simulation does not include the ECU in it. Instead the model is simulated with the actuator inputs from the ECU. This can be clear from the figure shown below:

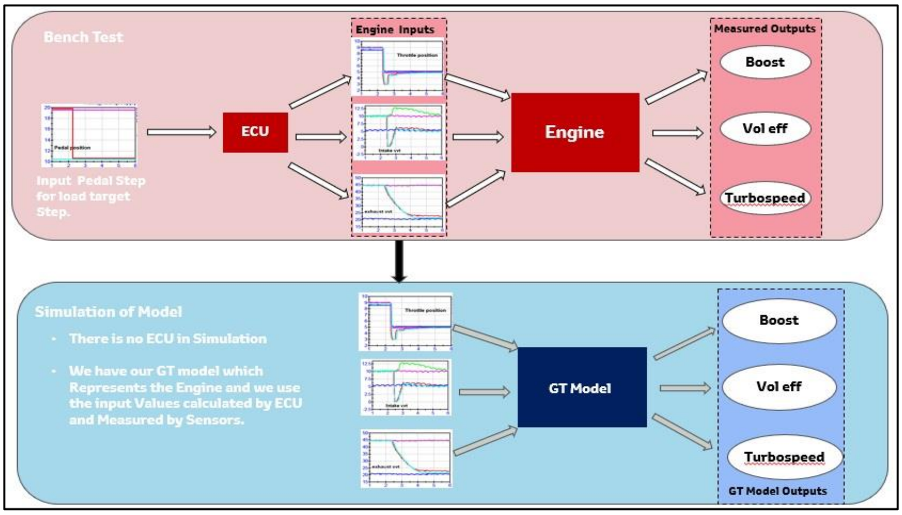


Figure 25- Bench Test vs Simulation Process

. The model should be simulated with the actuator inputs from the ECU such as :

- Throttle
- Wastegate
- Spark advance
- Injection timing
- Injection rate
- Variable valve Timing.

The model simulates the engine behavior using these inputs and predicts the output parameters such as

- Volumetric efficiency
- Boost pressure
- Compressor speed
- Intake Pressure
- Turbo Scroll pressure
- Back Pressure

The model needs to be simulated at both steady and transient conditions with all the corresponding test inputs we obtained from the test.

The simulation of a GT model requires the correct inputs from the test. Other factors also need to be considered such as the initialization, Time step ,solver method etc. for a good simulation result. The simulation setup and robustness of the model is important because the model might need to be improved or modified over consecutive simulations based on the results. This modifications to model should not result in lots of changes in the simulation setup which might lead to lot of time consuming changes.

6.1. Simulation Harness:

The above required need for a robust setup is the influence to develop a test harness for the simulation. As it can be understood that the simulation acquires ECU data from the tests and the models are simulated using these inputs, a setup has to created so that multiple models can be simulated with ease for the same set of input data. The objective of the Test harness is to create an robust interface between the Simulation inputs and the Models for this purpose.

In GT-suite the models can acquire the input data from multiple sources(such as excel data, .mat files ,ASCII files) directly into them. The problem with this setup is when we use different models(in our case we are using detailed and Mean value Model) for the same simulation inputs , we need to setup both the models separately and initialize them separately. This could be a time consuming process. And another important issue is while updating the models, as the setup procedure might have need to be done again while adding new components to the models.

This can be solved by using a harness model for the simulation setup. A harness is basically a main model where you give in all the inputs and initialize them. The engine model is a sub model inside the harness model where these inputs are signals connected to the sub-model. There is a wireless interface inside the engine model which transmits the signals to the right components. In this case we can use multiple engine models inside the same harness and simulate them with ease.

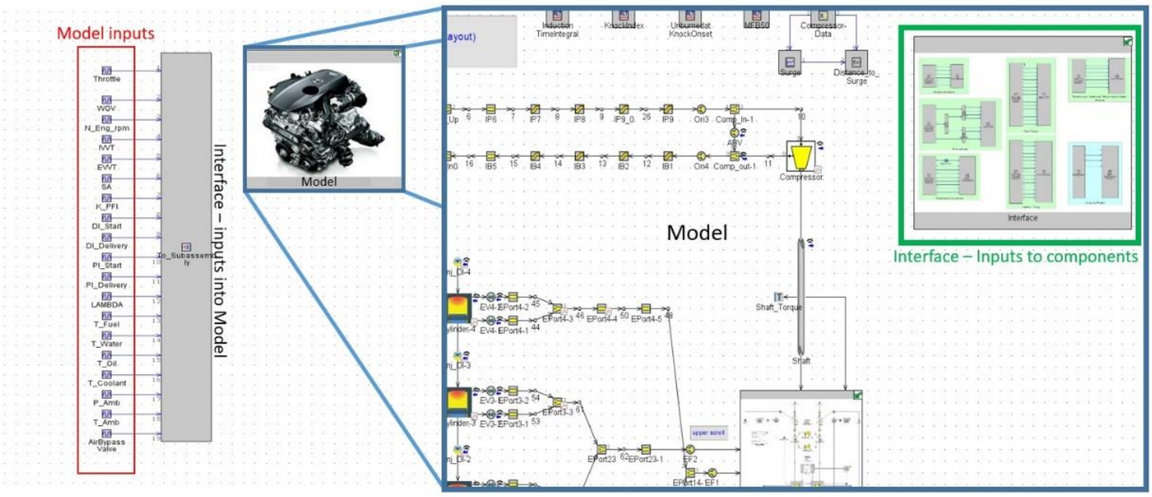


Figure 26 - Main Harness Setup

The figure 23 corresponds to the Main harness model where the inputs are provided into a wireless harness part which connects them into the subassembly. The subassembly has the engine models where the models can be switched depending on the simulation type. Inside the model there is interface to send the input values into the respective components. Example VVT and Spark advance should be sent into the cylinder model. The Injection Timing and rates should be sent to the injector models etc.

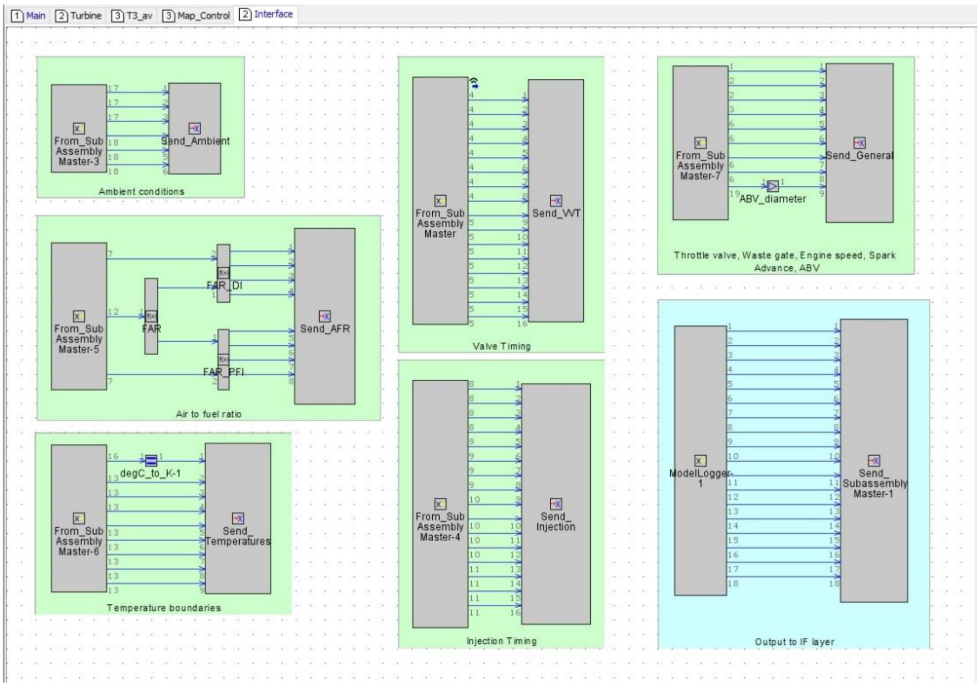


Figure 27 - Singals from Main harness to subassemblies of model

The advantages of using a harness setup summarized :

- Multiple models can be simulated for the same input conditions.
- Updated models can be just plugged in the subassembly .
- Controllers like PI can be easily implemented on the inputs if there are requirements. Using controllers inside models means making different models for different simulation setups. This can be understood in detail in the upcoming sections.

6.2. Steady Simulation Setup

The steady simulation of the model is simulating the model at the steady operating points. As it was previously seen that the engine was tested for 52 points over the operating range. The model can be simulated at these 52 speed-load points.

The follwing needs to be setup for the succesful simulation of the model at steady:

CASE setup:

The simulation inputs to the models such as VVT,SA,Wastegate angle etc are one single value corresponding to each operating point. These values were obtained from the steady testing of the engine. The simulation of every operating point is setup as a single case in the GT-suite with the imposed speed and the actuator inputs from the ECU.

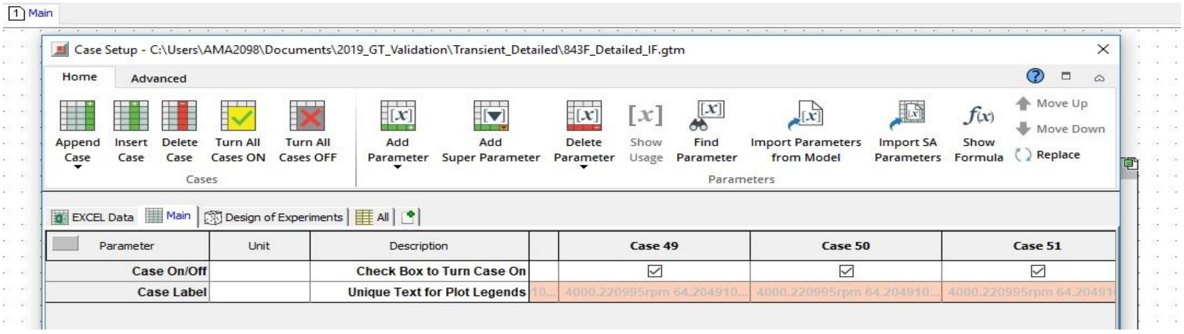


Figure 28 - Case Setup for steady Simulation

Though it's a steady simulation process the GT-suite runs every case in transient until it reaches steady state ouput values. The final steady state values are the model result.

Convergence criteria:

It is important for the simulation to converge to steady state as dast as possible. The initialtisation off the initial conditions determine how many cycles are needed for the outputs to converge to final value. Closer the initial conditions to the final value lesser the number of cycles to reach steady state and thus lower the simulation time. The initial conditions can be initalised from the outputs of the Steady test data . The convergence setup is done based on the tolerance over the cycles for the following outputs:

- Compressor speed
- Lambda
- Boost pressure
- Intake Pressure

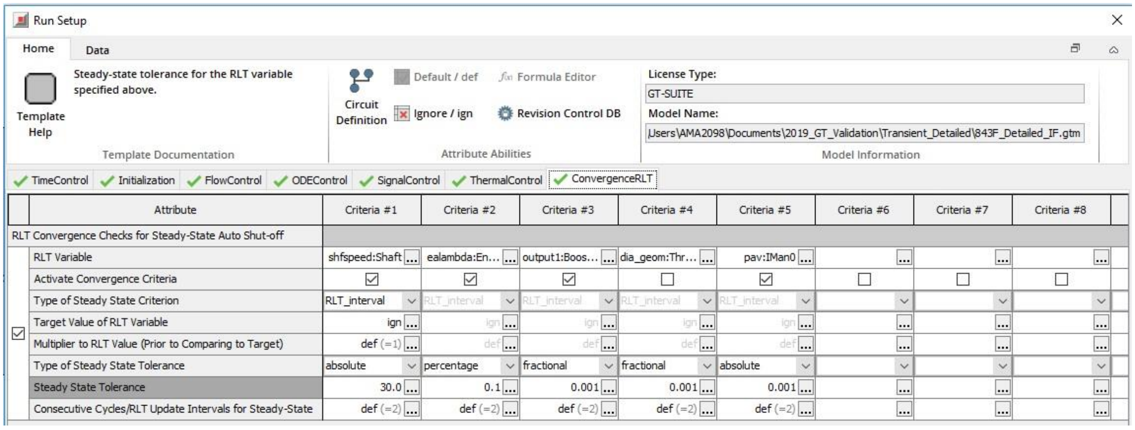


Figure 29 - Convergence Criteria imposed in model

Number of cycles(Duration time for each case):

Minimum and maximum number of cycles for every case should be determined so that the simulation doesn't run too long. It must be taken care of that the steady state value can be reached within this maximum duration. The option 'Automatic shut-off when steady state' should be turned on in the run setup as the simulation should end when steady state values are reached. This saves time.

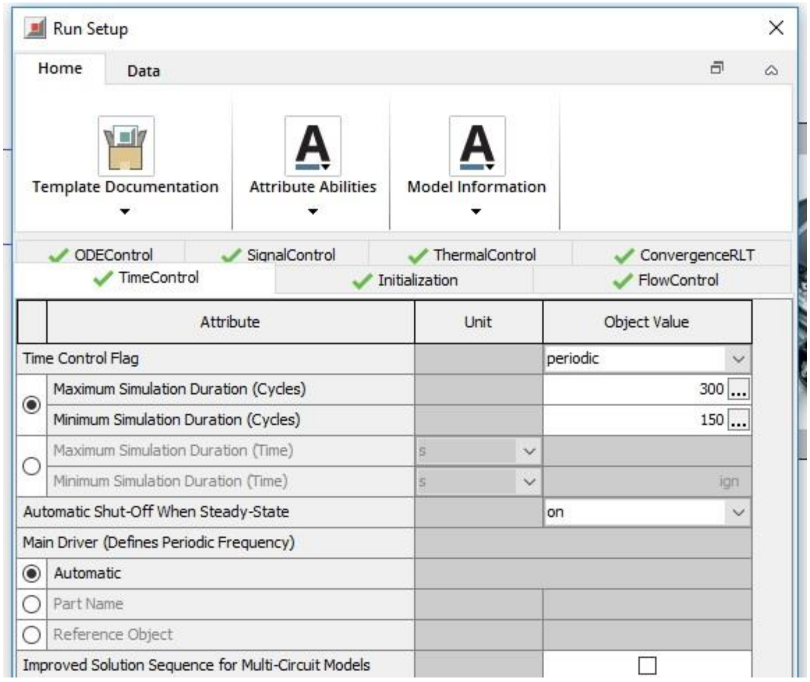


Figure 30 - Cycle duration and Automatic Shutoff

Closr loop setup for accurate results:

In the steady simulation it is essential to ensure that the model can predict the right outputs such as volumetric efficiency,Turbospeed,Scroll Presuures,Back Pressure , Intake pressure,Boost Pressure etc. Small inaccuracies in the Wastegate and Throttle models can cause large inaccuracies in the Boost and intake pressures predicted by the model. If the Boost pressure and intake pressure values are inaccurate it is impossible to get the right volumetric efficiency and all the other parameters such as tubospeed,Scroll pressure,back pressure will be inaccurate too. This can make the validation process meaningless.

To make sure the models behave closer to the engine, a PI controller is used to target the wastegate values and throttle angle values for the respective Intake and Boost pressures obtained from the steady testing. Since it is not a transient simulation we do not concern ourselves with the response of the actuators. All we need to make sure is that the model can reach the right boost pressure and intake pressure so all the other parameters predicted by the model such as massflow,volumetric efficiency,Scroll temperatures etc can be effectively used for validation.

The controller parameters of the PI controllers are tuned so that there is less overshoot and the target can be reached faster. Controllers with too agreesive propotional gain can lead to unstability and eventually the steady state will never be achieved or take too long for the model to reach. And if the integral gain is too low the time taked ot reach the set-point is too long.

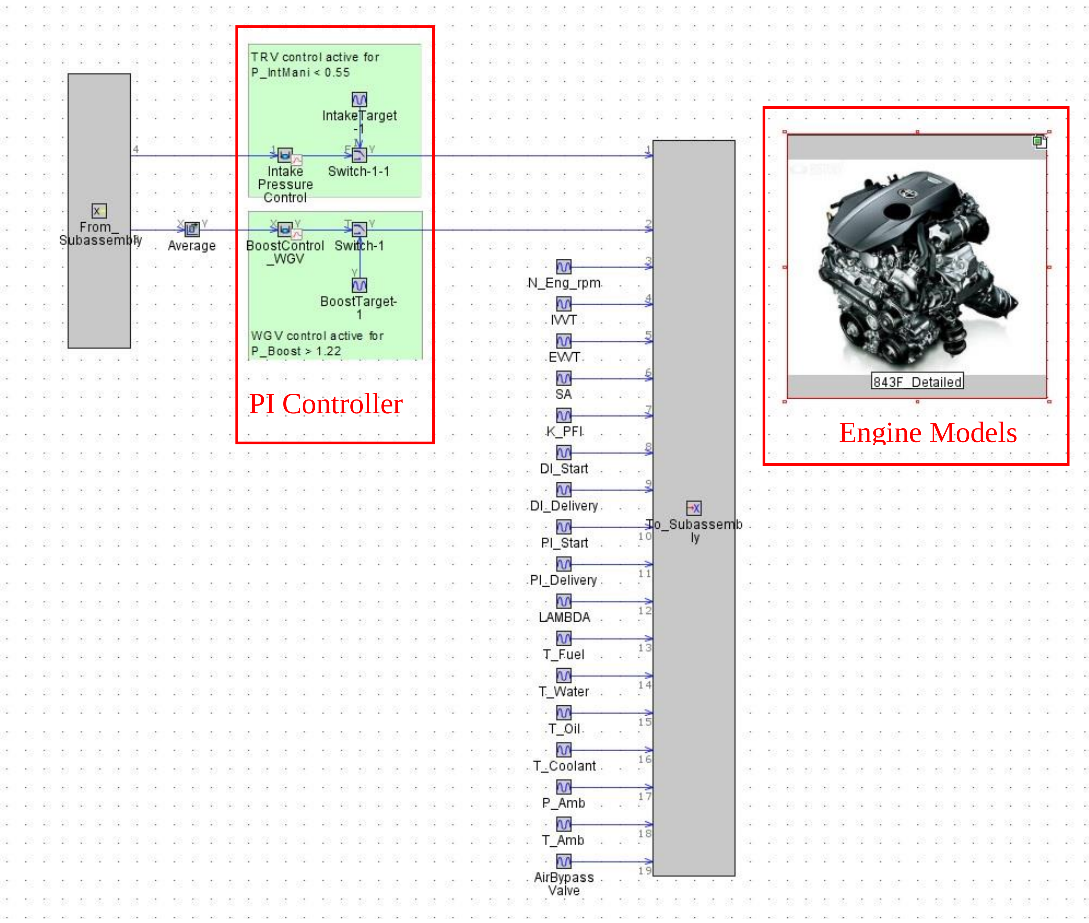


Figure 31 - Closed loop setup for Throttle and Wastegate

6.3. Transient Simulation

Tranisent simulation refers to simulation of the model under transient conditions or time dependant conditions. This is similar to the transient tests we performed where we obtained the outputs of the engine over time for actuator response. It is the same for a transient simultion where the actuator responses are given as inputs to the model over time and the model is simulated.

There are two running modes in transient. The speed mode and the load mode. The speed mode refers to imposing the speed and predicting the load. In the load mode the load is imposed and speed is predicted. In our case the simulation is done in speed mode as the test was performed as load step at constant speed. The simulation setup resembles the same setup as the test . The speed measured from the test is imposed in the simulation and the engine performance parameters such as Volumetric efficincy, Turbo speed,Boost pressure ,Intake pressure, Massflow etc are predicted by the model.

Input setup:

The inputs are defined as time dependant arrays using a ‘Profiletransient’ object in the GT-suite. The profile transient can be given the array data from the excel coloumn corresponding to the time. The profiles of VVT, Spark advance, injection timing, Wastegate angle ,Throttle angle , Speed etc are inputs into the model. This setup is shown in the figure below.

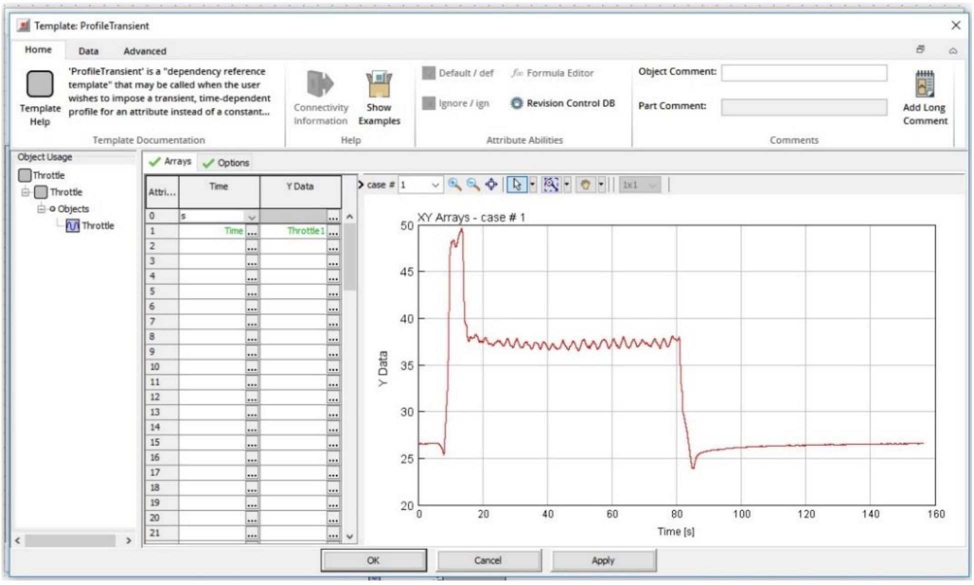


Figure 32 - Setup for "Profile Transient" object for transient simulation

Unlike the steady simulation case there are no controllers used here. This is because we are testing the response of the model in transient to the actuator response . The wastegate and the throttle are in openloop. The model uses the measured Wasgtegate and throttle angles from the tests.

Run Setup:

Thermal solver in the run setup has to be set to ‘TRANSIENT’ mode. Since we are performing a transient test the solver should not shut off when steady state is reached. So the option ‘Automatic shut-off when steady-state’ is switched off in the run setup.

Timestep of simultion:

As it’s a transient simulation the calculation timestep of the solver should be determined. The timestep is determined as a fraction of the crank angle degree. The timestep is important as it determines the calculation intreval of the solver. The more the time-step the less accurate the simulation is. And too less the Time-step the more computational time and load the model takes up. The time-step should be optimum. This is determined arbitarily by running some test simulations and checking the resolution of the results.

Process Automation:

Another major challenge faced during the transient simulation is the automation of the simulation process. In the case of the steady simulation we run 52 different cases which the GT-suite completes in one full simulation and the results are logged for all the cases. The case results can be obtained from a single file. In the case of transient simulation each simulation is a separate process. Means for every transient simulation the inputs are different and loaded from a different test excel file. It is very inconvenient to run each simulation manually over and over again due to the large number of transient data needed to be simulated.

This can be solved by automating the whole setup in using Matlab. The GT-suite can be called using a cmd code in the command window. And the GT results can be extracted using a command. Matlab has a cmd() function to run commands through it. A matlab script can be employed to automate the whole transient testing process.

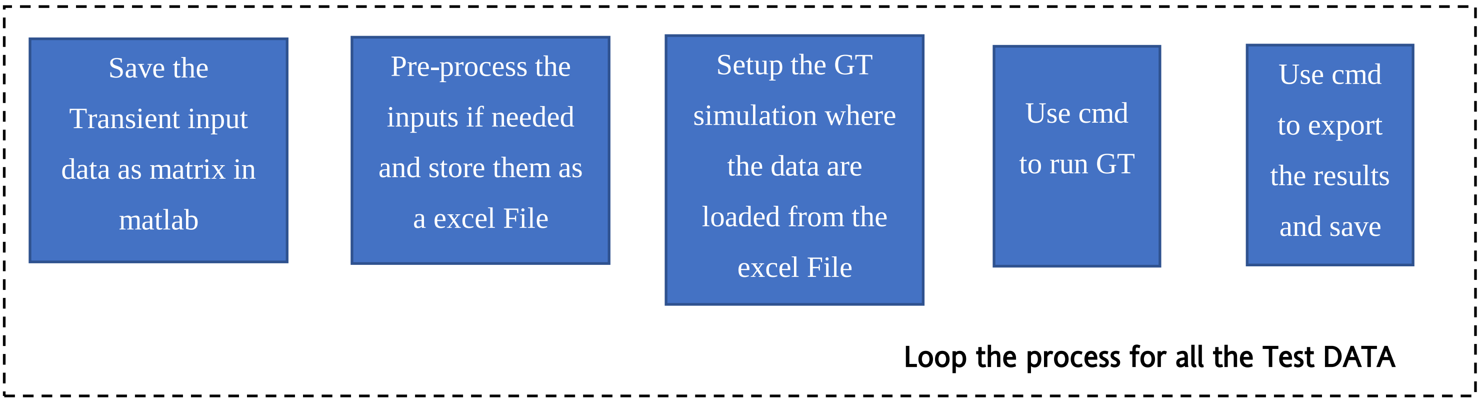


Figure 33 - Automation Process for transient Simulation

7. Post Processing of DATA:

Post processing refers to the processing of the data obtained from the simulation so they can be used as for validation. The purpose of post processing data is to visually analyze them by plotting the results.It is done in the following steps:

- Acquisition of data from the simulation
- Reading and plotting the simulation data against the test data

For the case of data acquisition from the model GT-Post can be used. The time dependant results can be viewed and anayzed using the GT-post. Gt-Post has inbuilt plot and data selection tools using which the results can be read. Or a export configuration file can be setup in gt-post and the data can be extracted into a .txt file. This .txt file can be processed in matlab later to analyze and plot the results.

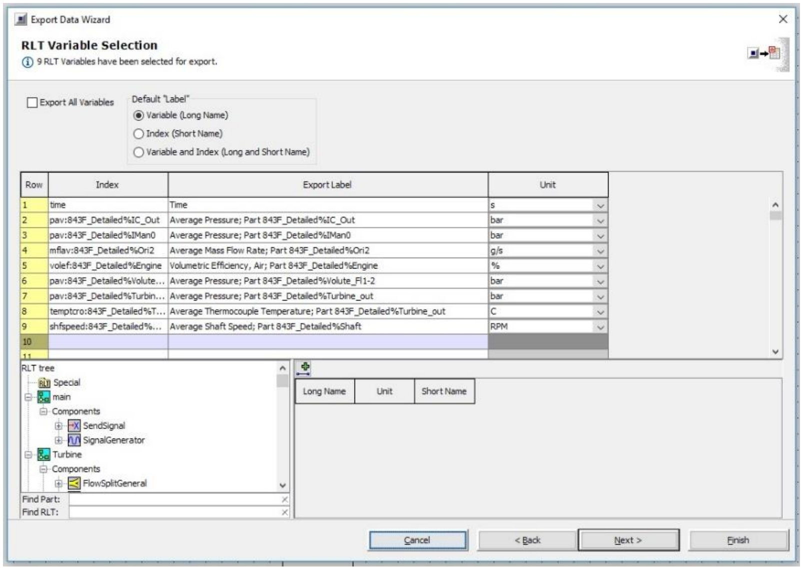


Figure 34 - GT Post export setup

As shown in the above figure the data that needed to be extracted from the respective parts can be chose and exported into a txt file containing the data in form of rows and columns. The data which has been saved as a .txt file can be read by matlab and saved as a matrix. The matrix data can be used to plot the simulation results against the test data for a few crucial parameters such as:

- Boost Pressure
- Volumetric Efficiency
- Intake Pressure
- Mass Flow
- Scroll Pressure
- Scroll Temperature
- Turbospeed

These parameters are considered as key outputs to analyze the behavior of the model when compared to the engine.

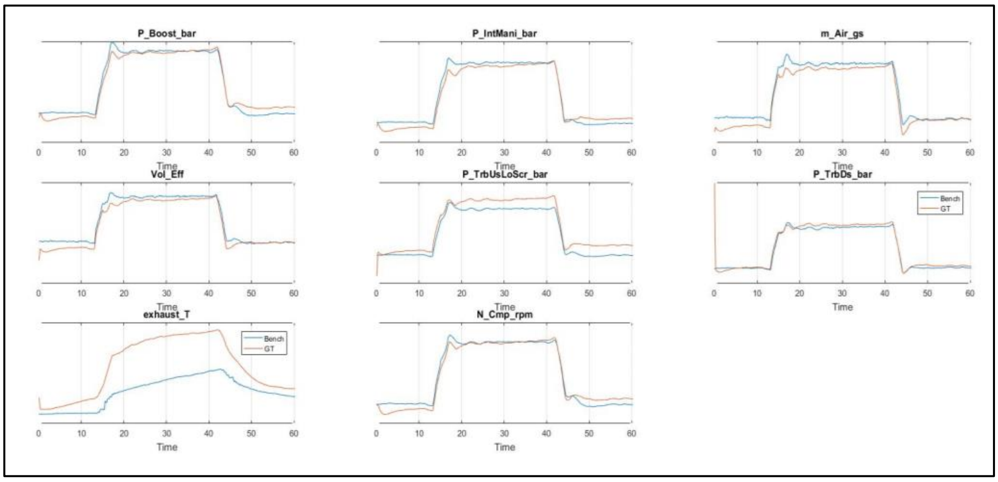


Figure 35-Test Bench data vs Gt-Model simulation data plotted

And the inputs from the ECU to the model and Engine is also plotted to analyze the behavior of the model based on the input values such as:

- Throttle valve angle
- Wastegate angle
- Intake VVT
- Exhaust VVT
- Spark advance
- Engine Speed

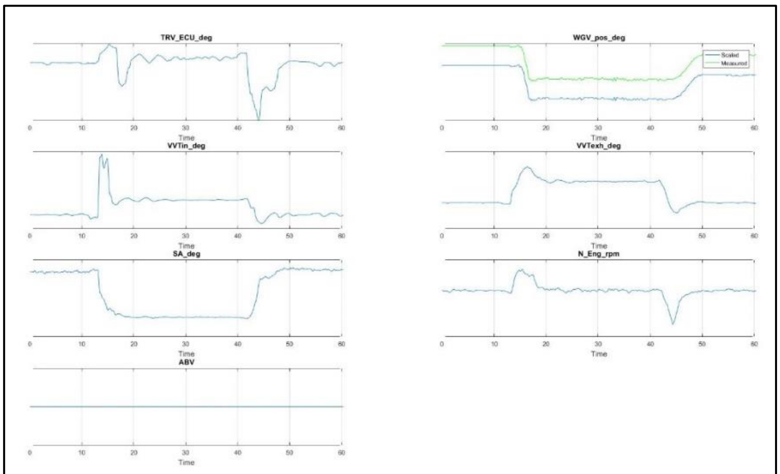


Figure 36 - Simulation inputs

8. Model Improvement

As it was previously seen from the process flow that model improvement is a loop in the process flow to improve the models based on initial validation. This is essential to ensure good model accuracy for obtaining the right results for model validation purposes. There we three places where issues were encountered or improvements was needed during the model improvement phase:

- Air Bypass valve
- Coefficient of discharge – angle model of the Wastegate vale
- Wastegate adjustment for model Accuracy

This section explains the about the above issues and how the model was improved to eliminate the issues.

8.1. Air Bypass Valve(ABV) investigation:

During the model checking/Initial validation process the model was tested against the engine conditions in transient conditions with previously collected step test data. The model was tested at a few step ups and down inputs to check if whether the model is good for the validation process. Analyzing the results it was found out that that at a few steps there was a very poor correlation of the model outputs to the test data. The problem and the improvement is summarized under the following sections.

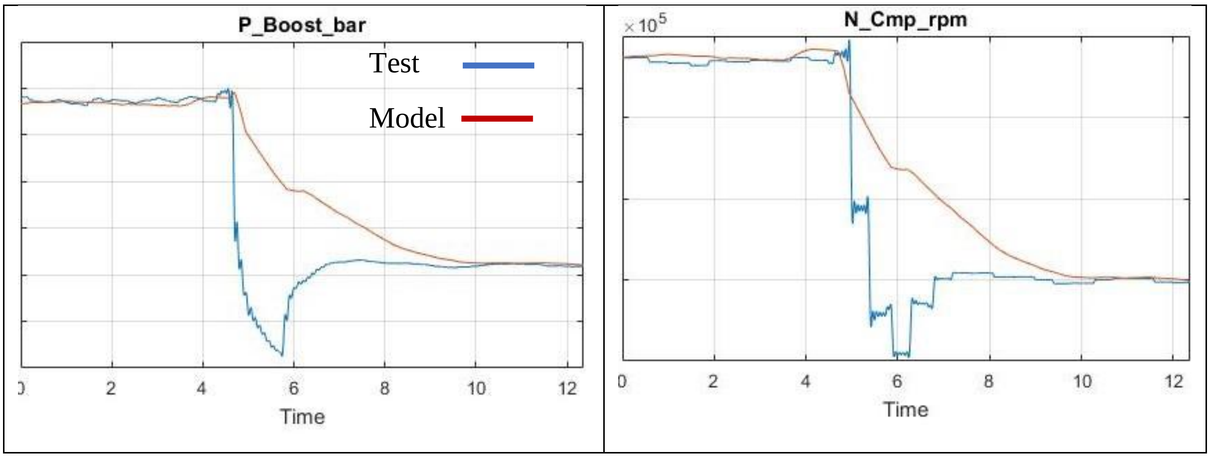


Figure 37 – Bad match of Boost Pressure and Compressor speed data from Test and Simulation

Issue:

As it can be noticed from the figure 33, during the step down phase the boost pressure and the compressor speed from the test data shown in blue has a sudden under shoot in the end of the step down before reaching the steady lower point. The value from the model on the other hand smoothly transitions to the lower point. This data shows that engine encounters an aggressive loss of Boost pressure and Compressor speed within a fraction of second which means there is a sudden blow off.

Investigation and Cause:

Upon investigation of the engine drawings it was found out that there is an Air bypass valve after the compressor. It is basically a valve which opens up into a pipe connecting to the compressor upstream and to the atmosphere eventually. The position of the bypass valve on the engine is shown by the figure below. It can be seen it is located over a pipe which runs parallel to the compressor in the engine airpath.

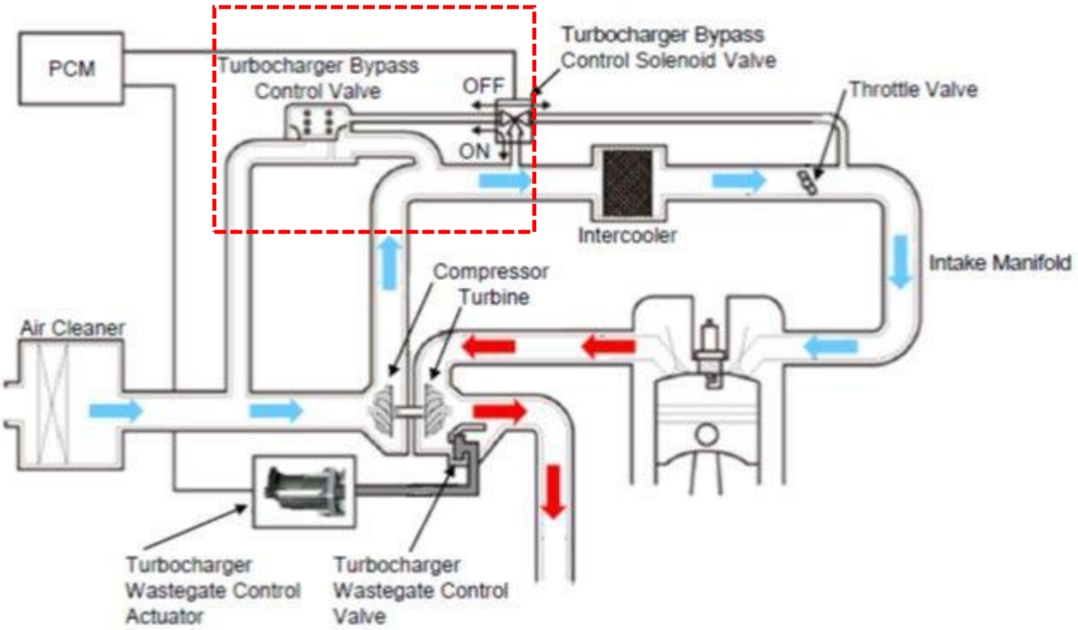


Figure 38 - Air Bypass Valve in the Engine

For a more brief in-depth explanation it's a surge control valve used to prevent compressor surge during a sudden tip out of the throttle. During a tip out and the throttle closes abruptly there is a high pressure buildup between the throttle flap and the compressor as the turbocharger is still being spooled by the exhaust gases from the cylinder. This causes a stagnation of air behind the throttle and the compressor blades cut through the air which can't flow. This causes a lot of vibrations in the compressor and can eventually damage the compressor bearings. The bypass valve can be either actuated by the vacuum from the intake manifold or electronically controlled based on a logic from ECU.

Countermeasure:

Upon Further investigation it was identified that there is a signal for the opening and closing from the ECU. This is a binary signal. To simulate the behavior of this valve a simple orifice was modelled between the inlet and the outlet pipes of the compressor in the model. The orifice diameter is controlled with the ABV signal recorded during the test. The diameter has a multiplier to the binary signal. So when the signal is one there is a positive non zero diameter value for the orifice which opens the flow across the orifice acting like a bypass valve.

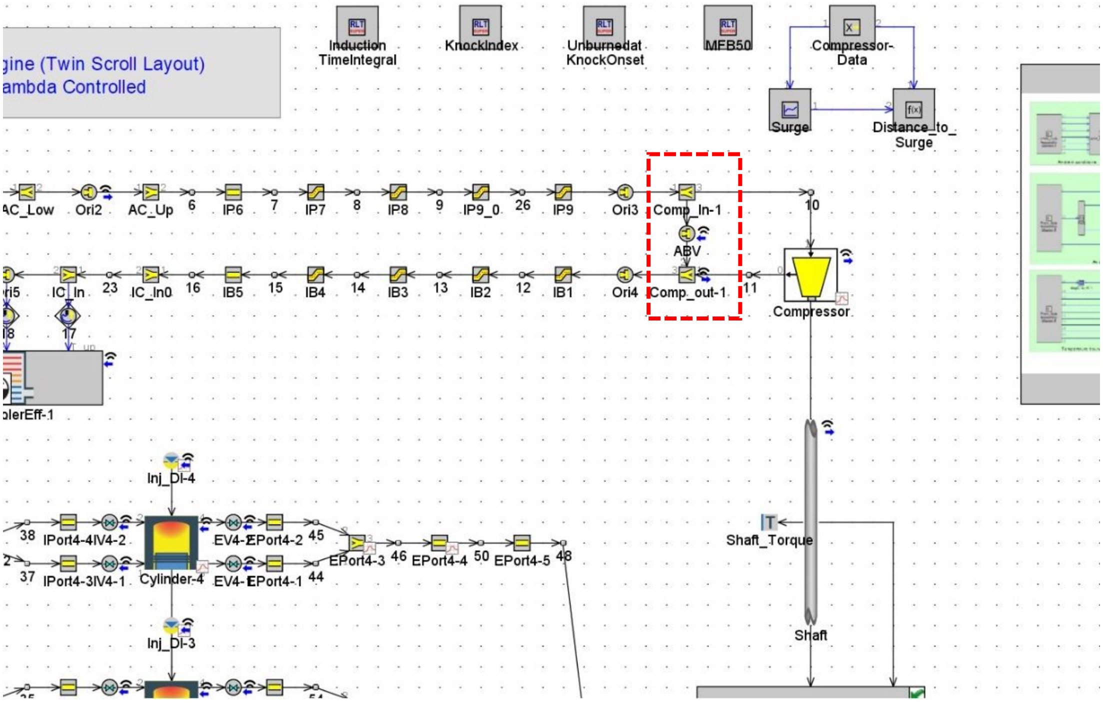


Figure 39- Air bypass valve implementation

From the above figure the ABV can be seen between the compressor inlet and outlet components.

Result:

After the ABV was modelled the model was again tested to check if can catch the engine behavior when ABV opens. Based on that, the diameter of the orifice was tuned so the model behaves as close as possible to the real engine in situations when ABV opens. Plotted below are the model-Engine correlation after the ABV was installed in the model.

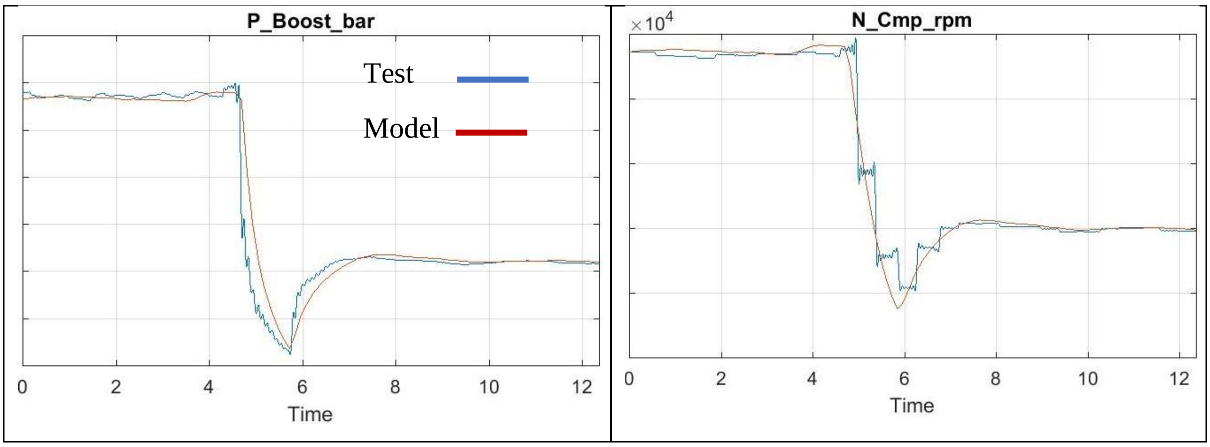


Figure 40 – Good match od DATA after model improvement

It can been seen now the boost pressure of the model has a good correlation with the Test data and is able to catch the undershoot when the ABV is opened. The compressor speed of the model also matches with the test data.

8.2. Cd-WGV model for Wastegate

Physically the Wastegate is a Swing Type Flap valve which can increase or decrease the flow across it based on the opening Angle. The flow across the Turbocharger Turbine is split into flow across the turbine and the wastegate. So if the wastegate is closed the flow across the turbine increases in-turn increasing the turbine Power. This leads to more Boost and higher volumetric efficiency.

To predict the right Boost pressure and the compressor speed by the model the flow across the wastegate should be modelled to simulate the right conditions when the wastegate is operated. Since the flow across the wastegate is our requirement, it can be modelled as an orifice where the flow across the orifice depends on the Coefficient of discharge, cross sectional area and the pressure difference against the wastegate.

$\text{Mass Flow} = \text{Cd} * \text{A} * \sqrt{2 * \rho * \Delta P}$	Equation 11

Cd - Discharge Coefficient

A - Cross Sectional Area

ρ - Density of Fluid

ΔP- Pressure Difference.

Assuming the cross sectional area to be constant we now have the dependency of the massflow on the Cd and the ΔP. But the Cd needs to be found out for every angle input of the valve to predict the right flow across the wastegate. This now makes the Cd dependent on the wastegate angle and the pressure ratio.

The steps to model the Cd – WGV dependency is given below:

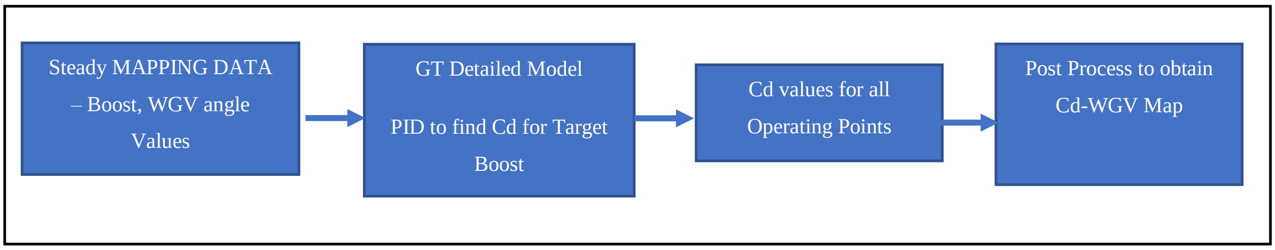


Figure 41- Cd-WGV angle model creation process

To model the Cd over the wastegate angle and pressure ratio, the Cd values for a combination of wastegate gate angles and pressure ratios needs to be obtained to create a correlation between them. This set of data can be obtained from the steady tests we performed over the operating points.

The detailed model can be used to find out the Cd values. This is done by simulating the model in steady condition with the wastegate angle from the steady testing. A PID controller on the Cd to target the boost pressure from the steady testing is used to obtain the Cd for every operating point corresponding to the wastegate angle and Pressure ratio at that point.

As the next step the pressure ratio from the model is plotted against the boost pressure. Here the pressure difference is the turbine upstream pressure and the pressure turbine downstream Only the points which are in the Boost zone are considered. To calculate a meaningful Cd the pressure difference must be high enough.. The pressure difference is significant only in the boost area.

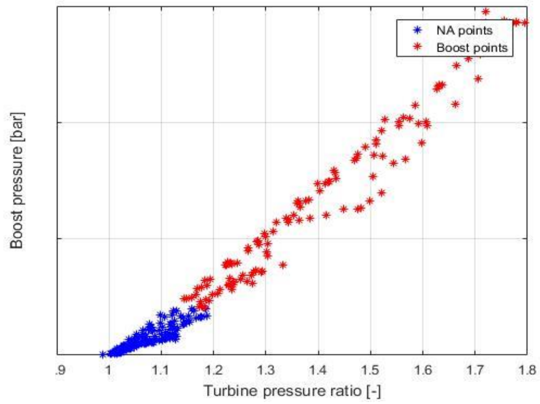


Figure 42 - Boost pressure vs Turbine Pressure ratio

The red points are in the Boost and considered for the Model creation.

The Cd-WGV relation is obtained by generating a line through the Boost points using least squares Regression Method. This method is used to find the best fit line which passes through a set of points by minimizing the square of the error between the actual point and fitted point.

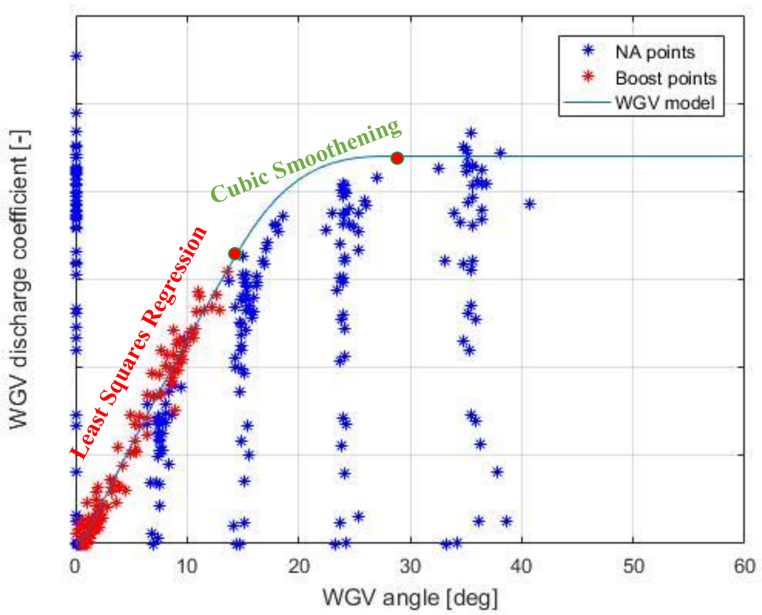


Figure 43 - Least squares regression for Best Fit line

Least squares regression[15]:

Looking at the trend of the Cd points it can be seen a quadratic curve can be used to fit the points to get the best relationship between the Cd and the angle.

Fit a quadratic line resembling the following equation over n points from i=1,2,3...

$y = ax^2 + bx + c$	Equation 12

In matrix form over determined linear model of the form of equation … can be expressed as

$y = X\beta$ + error	Equation 13

Which is expanded by :

$\begin{bmatrix} y_1 \\ y_2 \\ y_n \end{bmatrix} = \begin{bmatrix} {x_1}^2 & x_1 & 1 \\ {x_2}^2 & x_2 & 1 \\ {x_n}^2 & x_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$	Equation 14

The coefficients for the best fit line $\hat{\beta}$ can be figured out by solving the minimization problem

$\hat{\beta} = arg_{\beta}\,\mathrm{min}\lVert y - \ X\beta\ \rVert^2$	Equation 15

The solution to the above problem is given by [15]

$\hat{\beta} = (XX^T)^{-1}X^T y$	Equation 16

This method can be used to model the cluster of Cd points obtained targeting the right boost pressure using the appropriate wastegate angle from the test data. A best fit line is fitted through the cluster of points in the boost area to model the Cd-WGV angle relation . The Cd after a certain angle can be assumed constant as the engine doesn't operate in boost conditions at this area . A cubic fitting line is used to smoothen the curve between the regression line and the constant line. This 2D map is used as a Lookup table in the WGV model to determine the Cd of the Valve for wastegate angles.

8.3. Adjustment of Waste Gate angle value:

During the initial model validation/checking phase the model was simulated to analyze the behavior and it was encountered that the model cannot reach the outputs from the Bench Test during the transient simulation. The model could be seen responding well to the time based dynamics such as the rise and fall but could not reach the steady values from the test data and a shift could be seen from the figure.

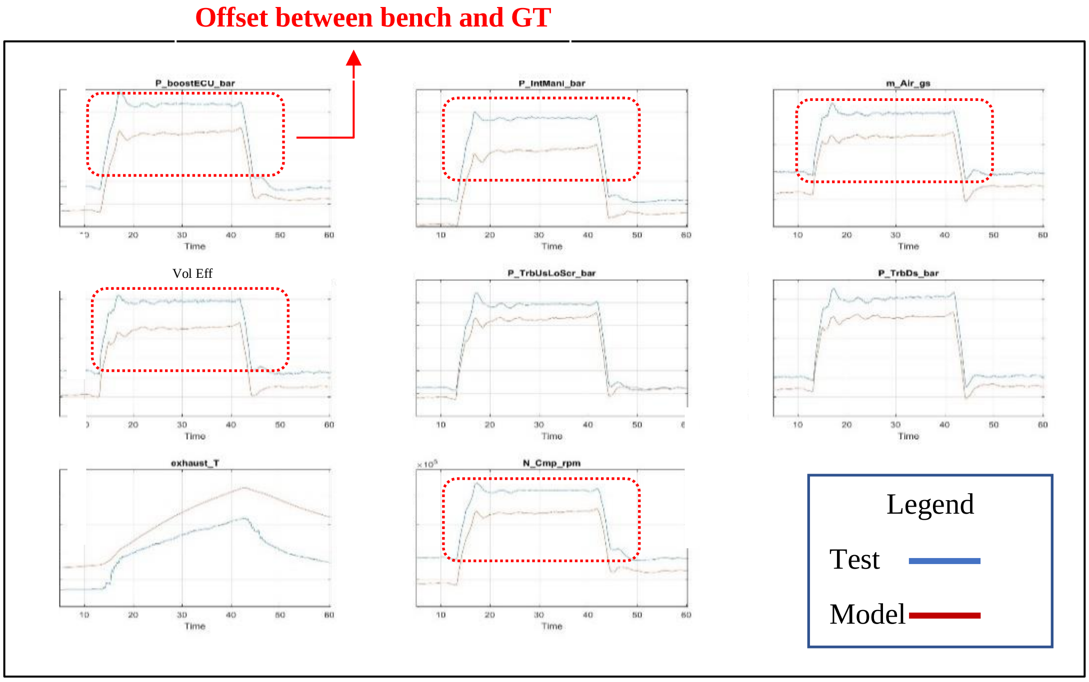


Figure 44 - Bad match between test and simulation data at steady zone

Since we need the model to be as close as to the actual engine for effective validation purposes it is required to fix this problem.

Investigation and Cause:

Since it can be seen that there is a shift in the data it could mean the model cannot reach the right boost with the actuator inputs. The most probable cause could be the wastegate valve. Though we made the wastegate model with the Cd obtained to match the wastegate angles from steady simulation, the actuators can have some fluctuations even at steady state and may deviate a bit around. This can affect the wastegate model as it can be quite sensitive to the change in angles.

The sensitivity of the wastegate model which it characterized by the Cd-WGV correlation can be explained as follows. Since the Cd curve is a best fit line created using line fitting by least squares method over a cluster of points we do not actually obtain the actual required Coefficient of discharge. Instead we obtain an approximate Cd value close to the actual Cd point obtained to match the right boost pressure using the PID. Since the slope of the Cd-WGV curve shown below is large and a small change in the wastegate measurement can cause the Cd to shift drastically and resulting a higher error compared to the actual required Cd. This sensitivity can lead to a error in the wastegate model leading to a large error in compressor speed and boost pressure eventually affecting the whole model.

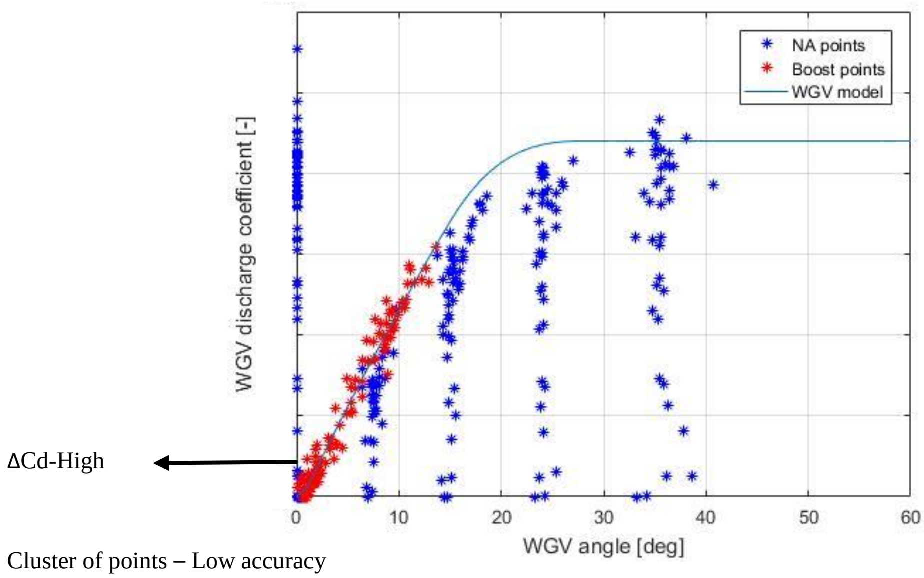
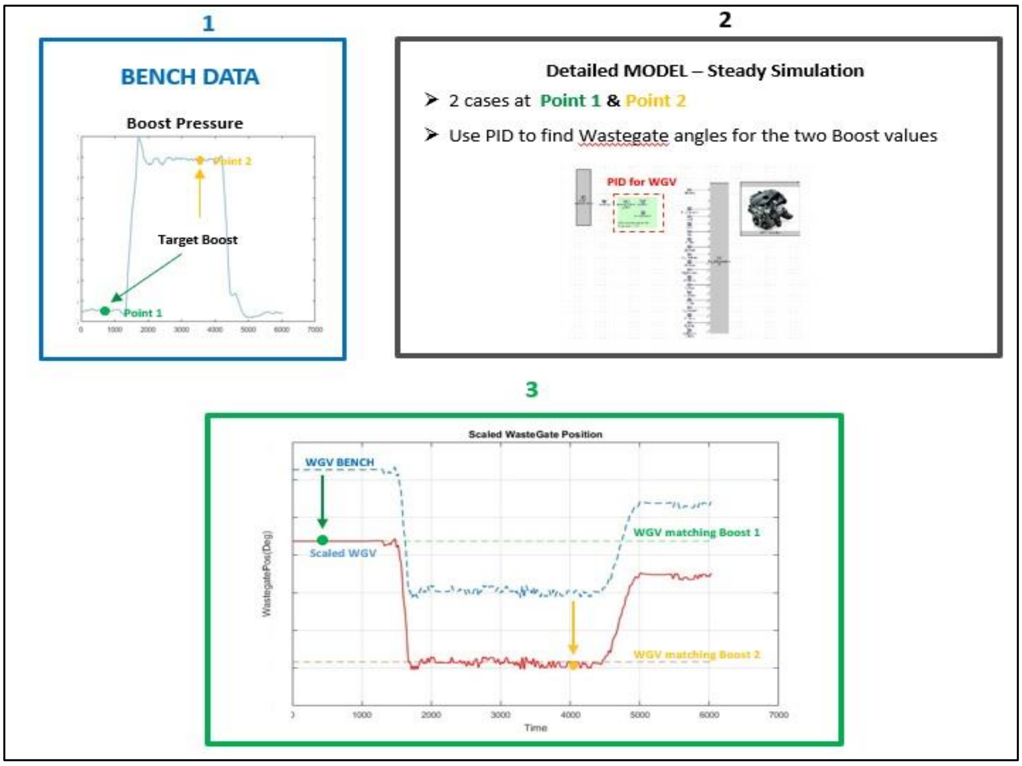


Figure 45 - Cd-WGV model Inaccuracy

Countermeasure:

To make the model match the actual boost from the engine and still keep the dynamics of the Transient , adjusting the wastegate data from measurement to match the Boost is done. This is to compensate for the error in measurement and the sensitivity of the model. The adjustment is done with the following procedure also shown in the scheme below.



- Two points at the high and low parts of the step are considered.

- The two points are basically steady data as the area they are in is at steady state before and after the step. The model is simulated at steady conditions with a PI controller to find out the wastegate angle corresponding to match the Boost pressure from the test. This gives us the wastegate angle values at the two steady points we simulated which is capable of making the model reach the right boost pressure.
- As we can see in the third step the scaled values are lower than the actual values at the two points. A linear transformation is done to fit the data to pass through these two newly obtained data. By this way we do not lose the transient behavior of the step but now we can be sure the new data can help the model match the steady parts of the transient step.

To make sure the wastegate values we obtained through the steady simulation are acceptable they are compared with the WGV values obtained during the steady simulation of all the points over the operating range. Since the simulation of the steady mapping points was done using a PI controller to find wastegate angle to match the boost pressure, the two angle values can be compared. A regression analysis can be done to compare the results between bench and simulation for the wastegate angles obtained during steady simulation and the wastegate angles values obtained during scaling process.

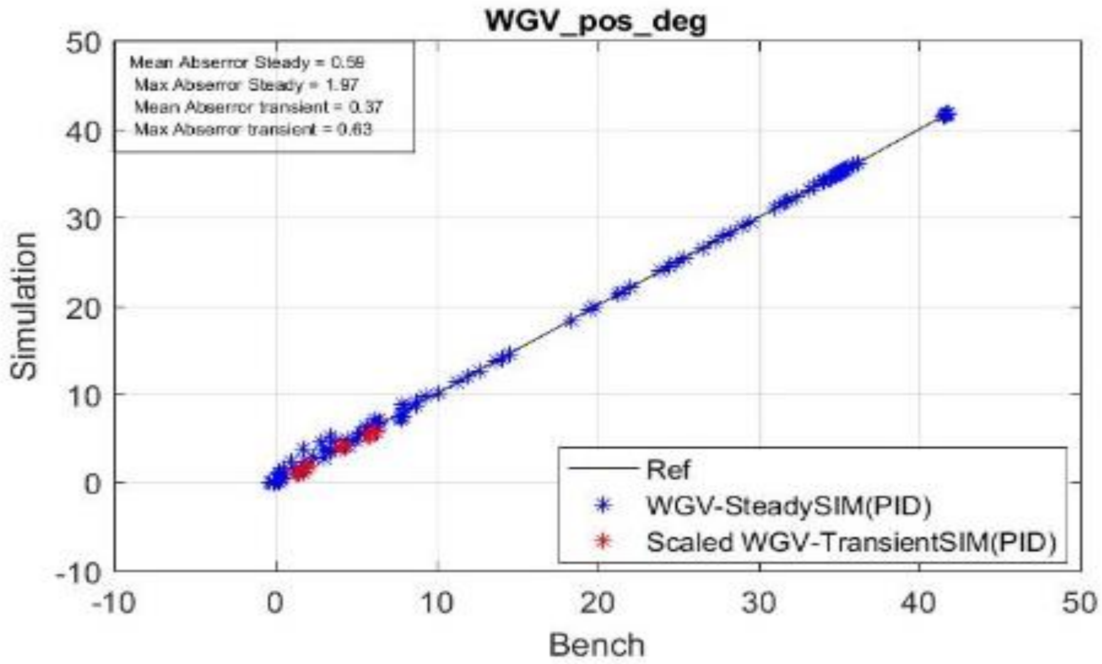


Figure 46- Steady WGV values from adjustment - WGV values from Steady simulation

We can observe from the results that the red points which are the scaled WGV values lie close to the simulated WGV values(blue points) from the steady simulation. So the points are reliable to be used for the scaling process.

Results:

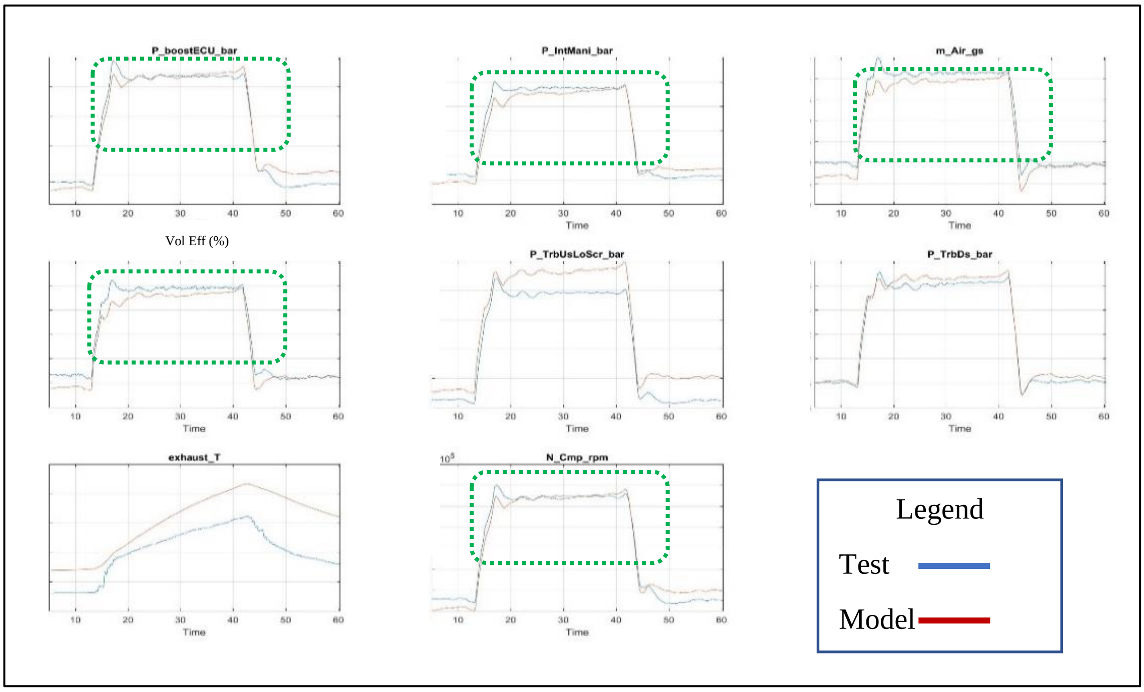


Figure 47 - Test vs Simulation after adjustment of WGV

After the scaling process the new Wastegate values are used in the GT model for the transient simulation of the model. This was checked by using the scaled wastegate values and the results were found to be good. It can be seen that now the Boost pressure from model is able to match the Boost produced by the Engine and the data is easier to validate since it is comparable.

9. Validation Methodology and Results:

Once the simulation is done and the results are obtained the results need to be compared with the test data for the validation of the models. For the validation to be done it is needed to:

- Devise a Proper validation methodology
- Determine the right Key performance index to compare the simulation data to the test data.

By keeping in mind the above two points the right validation method is created to compare the models.

9.1. Steady Mapping Points

The Engine is tested over the operation range for many different Volumetric efficiencies and Speed to characterize the performance. This data is used to validate the model. The model was simulated at these steady operating points, the steady simulation results can be used to validate the model compared to the Test data.

The objective of the validation is to determine Key performance index parameters to compare the data. As the steady testing and simulation data is a set of parameter(Boost Pressure, Vol-eff, Intake Pressure etc.) values for every operating point. A regression Validation method can be employed for out comparison. This data can be plotted as a regression plot and compared. Regression validation refers to plotting two independent set of data against each other and analyzing a linear best fit line passing over the points.

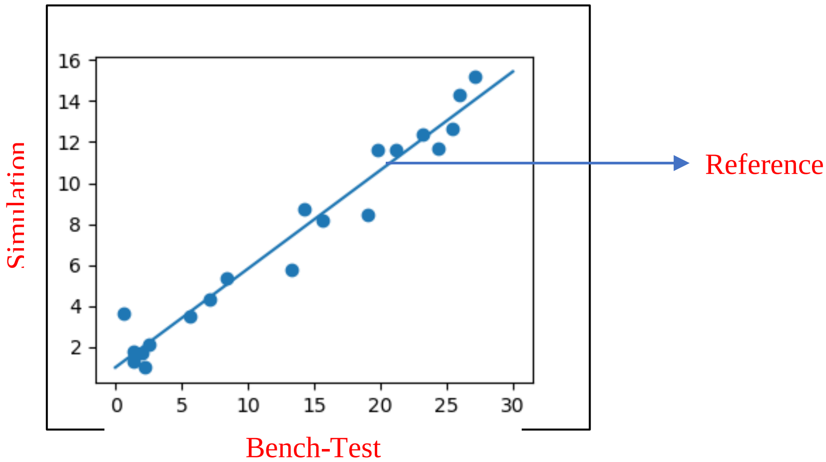


Figure 48- Linear Regression Validation

Many different analysis methods can be used to compare a regression plot. Since we are having 52 points to compare through a linear regression plot spread out over the reference line it is possible to compare the data using the average absolute deviation and the maximum deviation.

From the above we can determine the Key performance index for the validation are

- Average absolute deviation : This is the average of the absolute difference between each point and the reference value. The absolute is used because the difference can be positive or negative depending whether the point lies above or below the reference. Hence if we do not use absolute deviation we can end up having no deviation or very small deviation which can be inaccurate for our purpose of analysis. This helps us to analyze the spread of the data. It is given by the formula:
  $Avg\ Absolute\ Deviation = \sum_{i=1}^{n} \left| \frac{y_i - x_i}{n} \right|$ where n is the total number of observations and x is the reference value and y is the predicted value.
- Maximum Deviation: It is the point which has the maximum error with the reference data. This is more meaningful to see if there are any outliner points in the data as they can have an impact on calculating the spread. For example if we have a very few outliners with a very high error they can be identified and seen if they are significant or not.

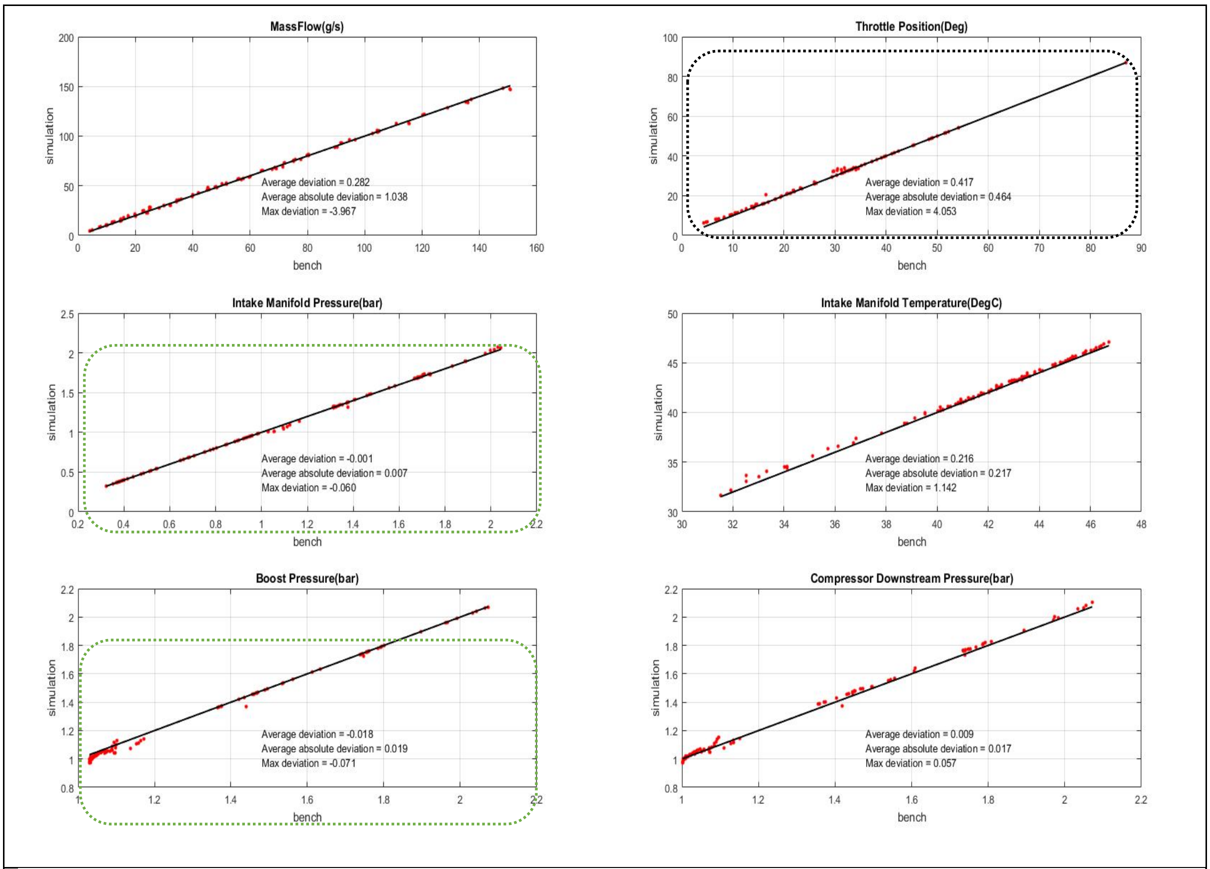
This analysis is done to compare two sets of data:

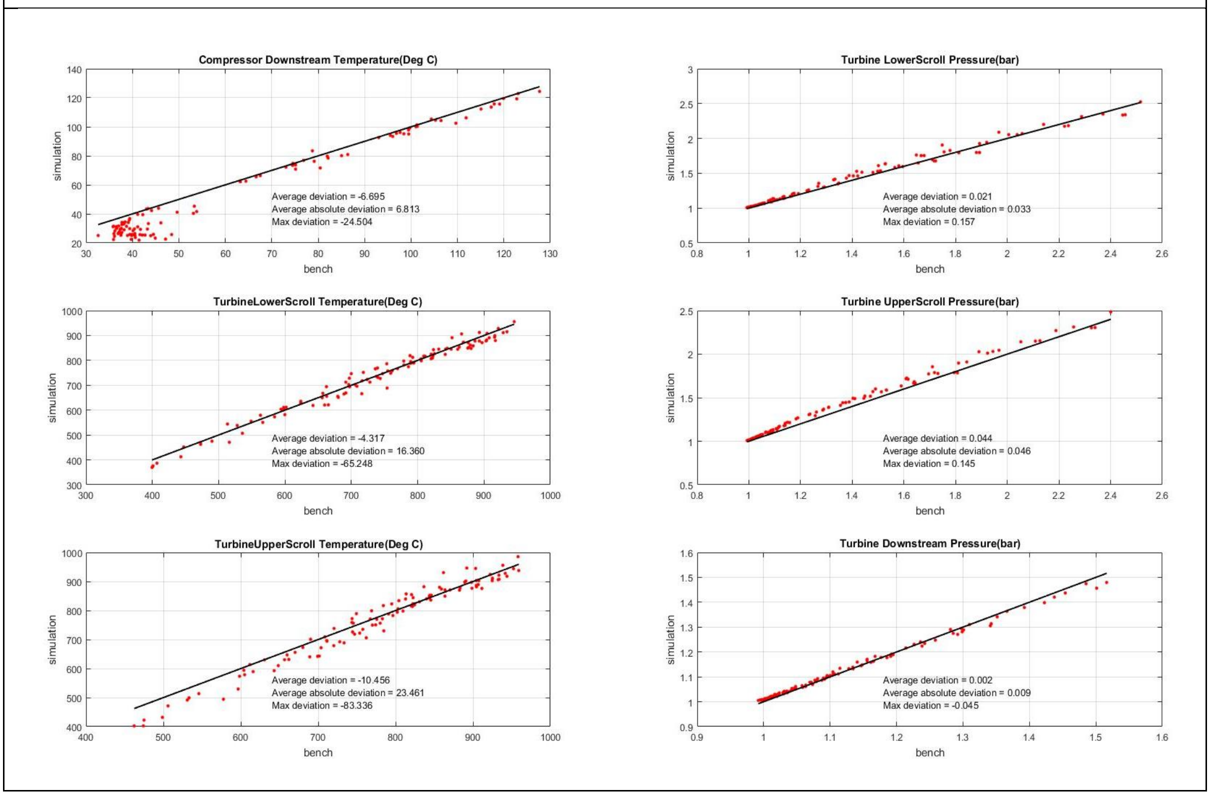
- GT-Detailed Model – Bench Test
- GT-MVEM – GT Detailed

9.1.1. Analysis and Results:

The results of the models against the test data are plotted as a linear regression plots to see the correlation.

GT-Detailed Model – Bench Test





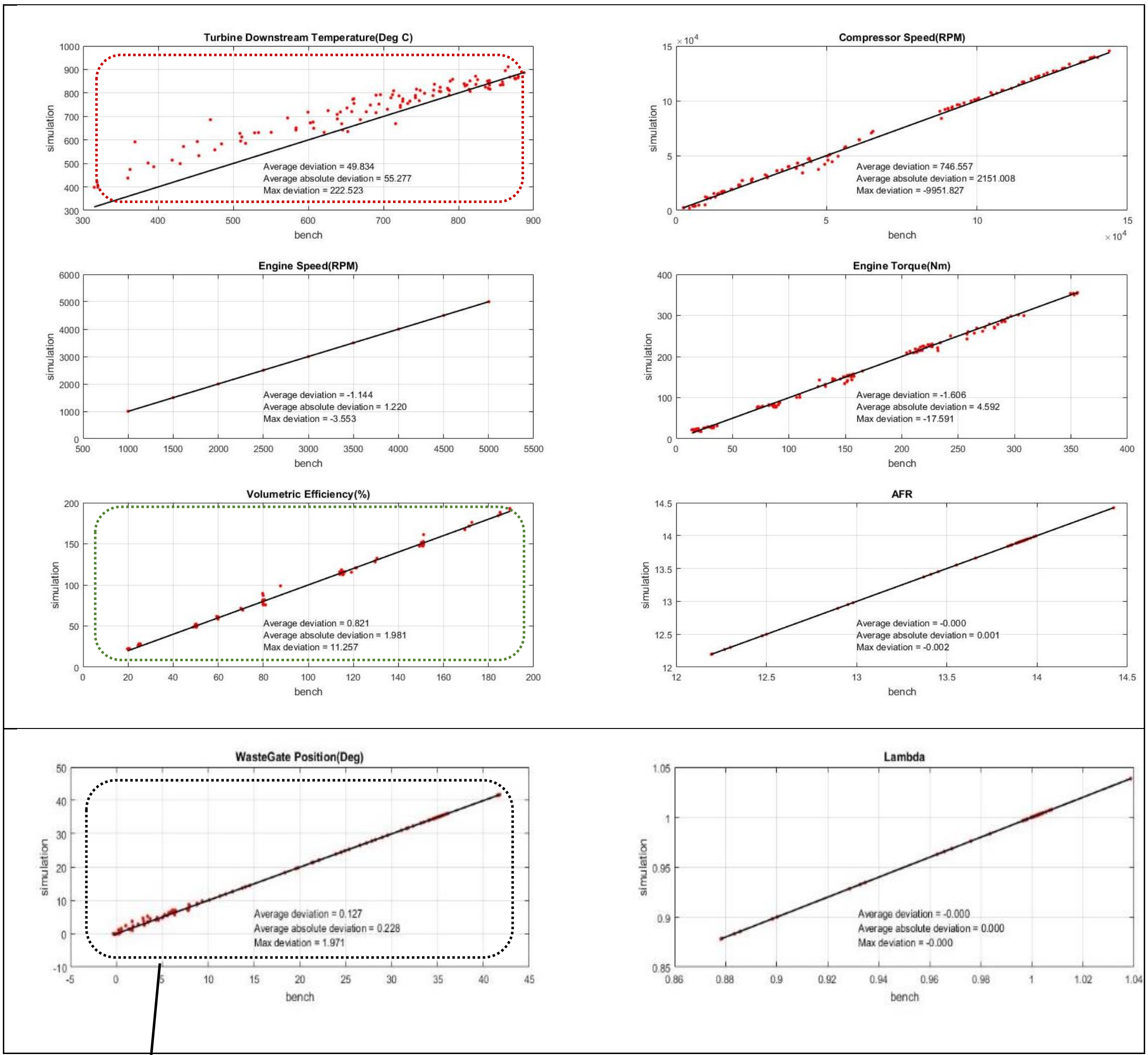
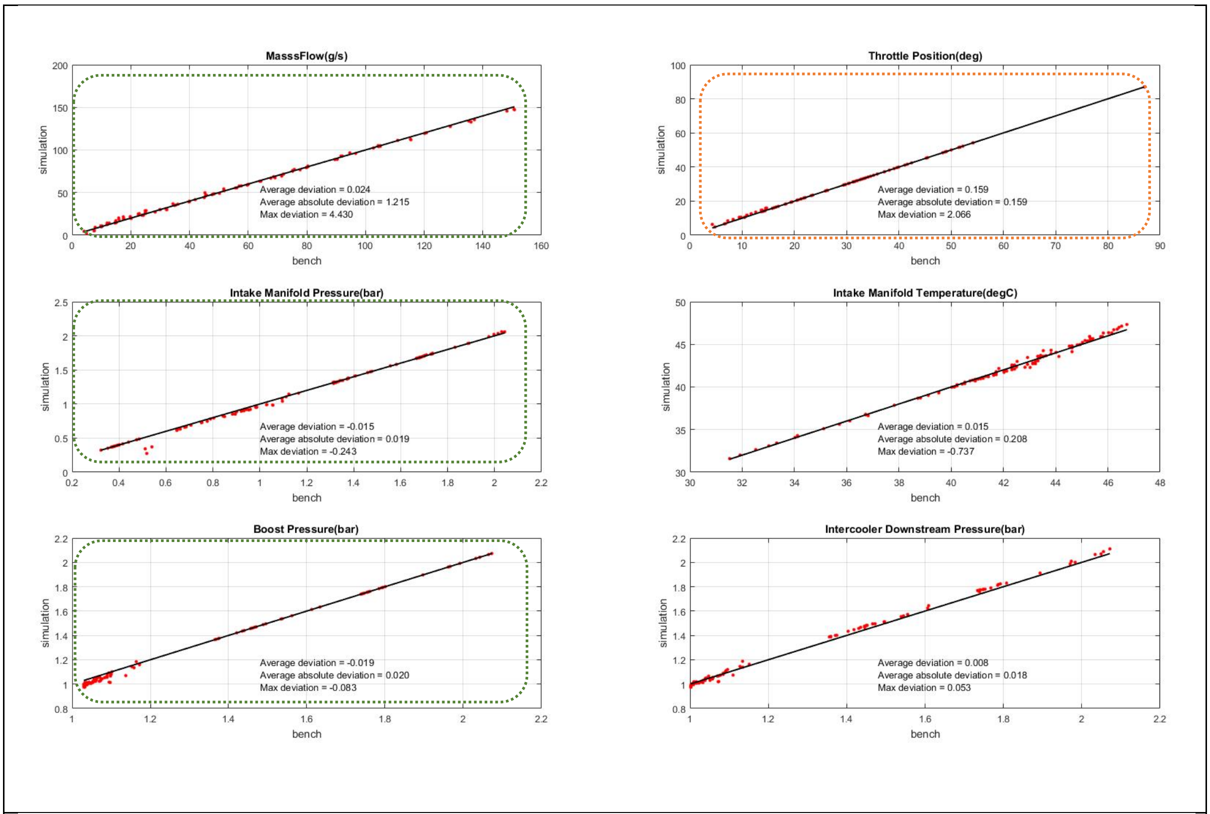
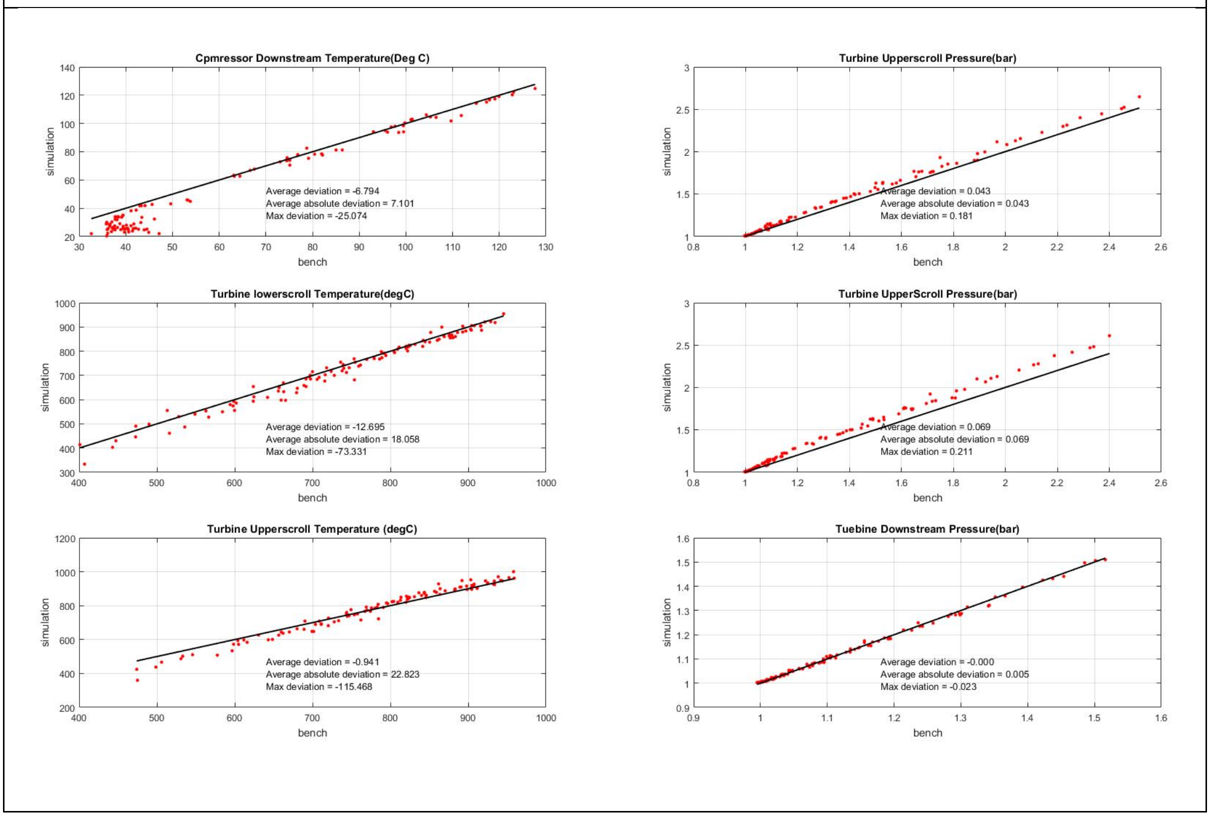


Figure 49 - Steady simulation results - Test vs Detailed model

WGV angle calculated by PI in model vs Actual WGV angle

Bench Test - Mean value Model Results:





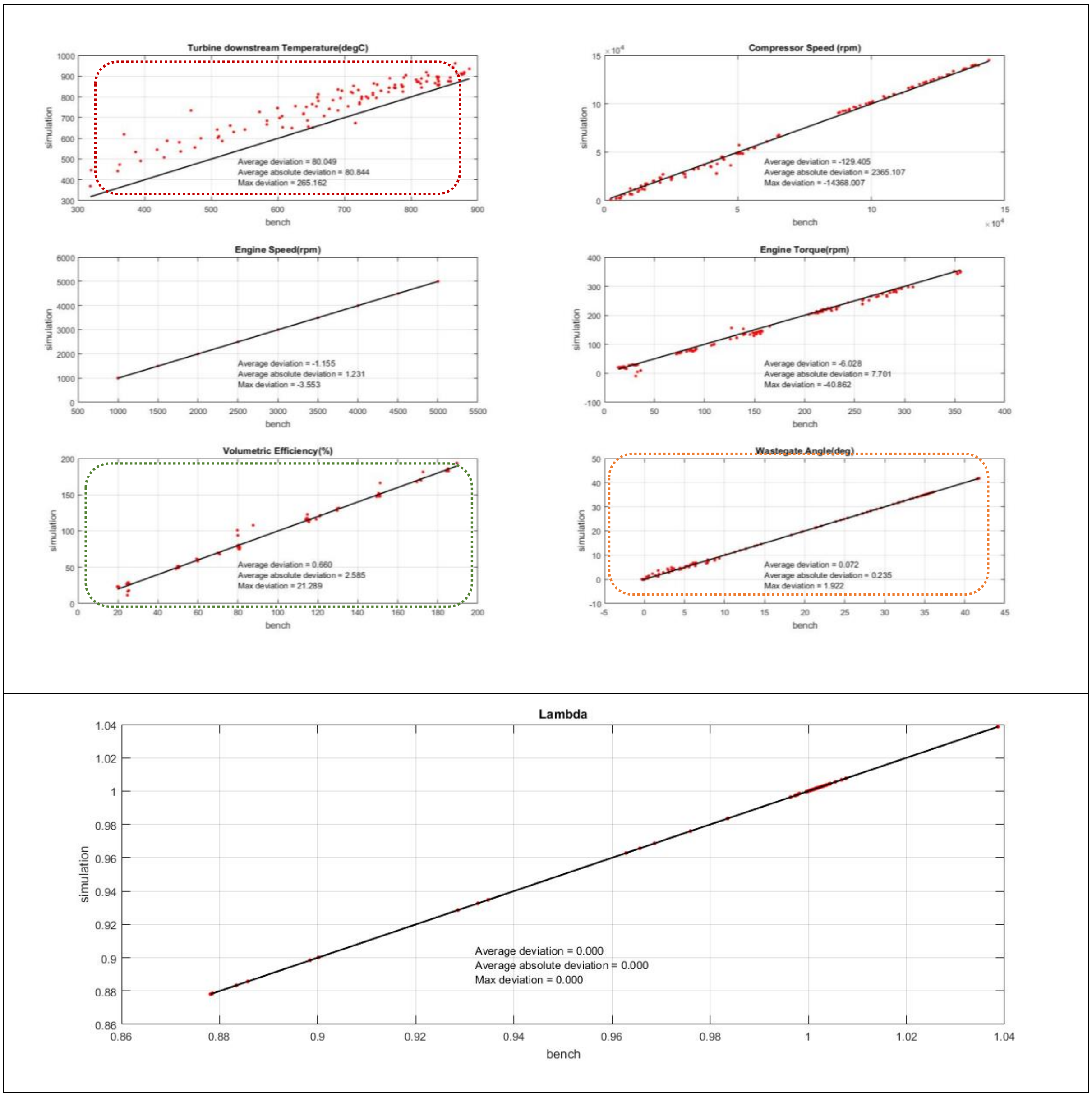


Figure 50 - Steady simulation results - Test vs MVEM results

Summary of results:

- From observing the results it can be observed there is good correlation to the test and the simulation in most of the results.
- A considerable error is observed on the Turbine downstream temperature.
- The Actual throttle and wastegate values are observed to be close to the values calculated by PI-controller in closed loop.
- The models are very good at predicting the Boost pressure, Intake Manifold Pressure, Volumetric efficiency and Massflow.
- Both the Detailed and MVEM models show similar behavior at steady at predicting the parameters.

9.2. Transient Step Response:

Recollecting the overview of the Transient Tests it could be seen that the Transient Testing was performed as a load STEP at constant speed. The same step input data was used to simulate the models in transient conditions. A methodology to validate the time dependent data is needed for the validation of the transient model behavior. We give more emphasis to the time response of the model as the model needs to predict the values over time correctly .As we can see from the test results the parameters (boost, massflow etc. resembles a second order STEP RESPONSE. The characteristics of a second order step response can be used to derive KPIs for the validation of the model and the Engine.

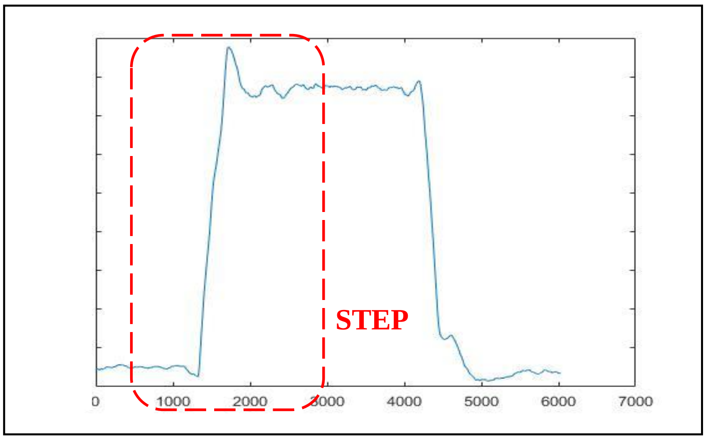


Figure 51- Transient step response observed from Test data

9.2.1. KPI Identification Method:

A Step response is a time dependent behavior of the output when a step input is given to the actuator. In a second order step there is a rise and then damped oscillations o the output before reaching the steady state. This behavior helps us to characterize the step response using the following parameters.

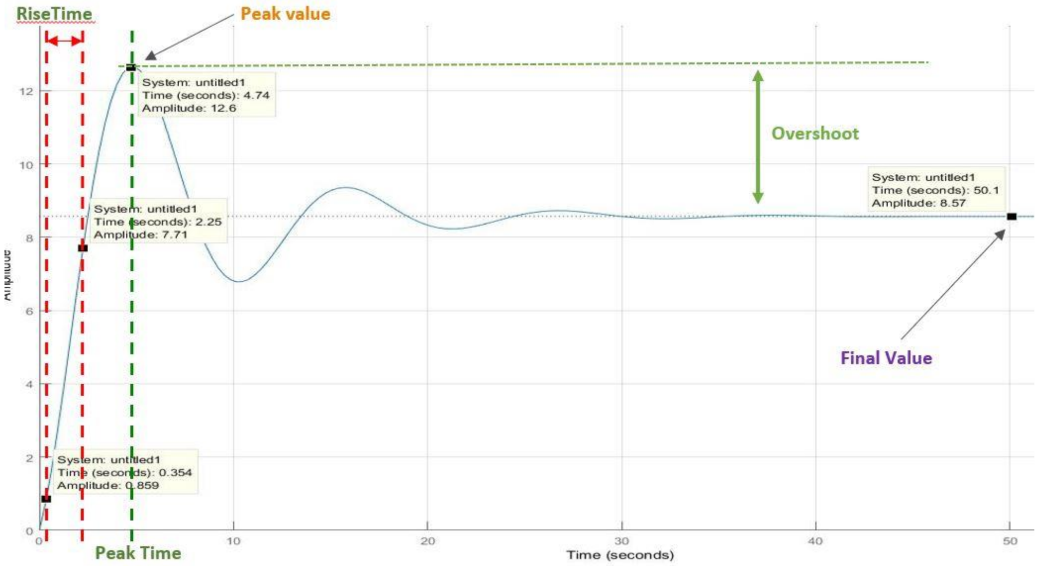


Figure 52- Characteristics of a Second order step response

- RiseTime – It's the Time taken for the step response to rise from the initial steady value to the Final steady value. Depending on the data and requirements of the user the boundaries of the RiseTime can be assumed by the user.
- PeakTime: The Time at which the Peak of the data occurs.
- Peak Value: The value of data at the Peak or the maximum value of the data.
- Overshoot: The percentage difference between the peak and the final settled value of the data. It can be explained as the percentage increase of the data over the final value before settling down.

Now that the data can be considered as a second order step response and the characteristic parameters of a second order step are explained, the suitable KPIs for the validation must be identified. It must be analysed and determined which of the above parameters can be used as KPIs for the validation of the model result. And also a reliable methodology should be determined to find the KPIs from the data.

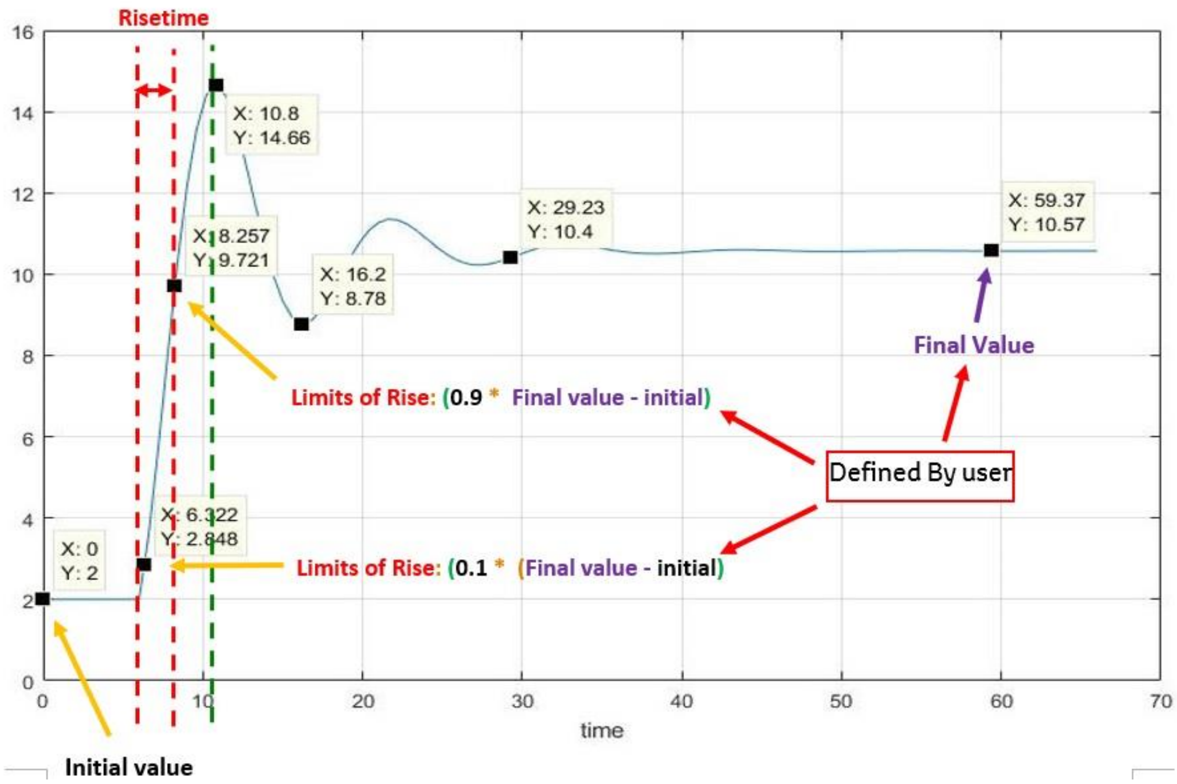


Figure 53 - Step characteriscs as calculated by stepinfo( ) function

A inbuilt function matlab can be used to find the RiseTime and PeakTime of a step data. It's called the STEPINFO( ) function. The function determines the RiseTime , PeakTime, Overshoot and Peakvalue of a step response. It uses the data value and time values to determine the parameters. The final value is set manually at the point the data has reached a steady value(this is done manually to be more accurate). This function is an effective method because it reduces the human effort and the time needed to write a code for the validation process manually. But before we apply the function we need to check if it can be used effectively for our purpose.

The function is checked if it can be used for our validation by simulating it with a second order step response resembling the characteristics of the data we have. The process is as follows

- A simple 2^{nd} order step response resembling our data is created.
- The characteristics of the created step response such as the RiseTime, PeakTime ,overshoot etc. can be identified from the graph from visual observation.
- The function is then applied on the same 2^{nd} order step and the results calculated by the function is obtained.
- If the function effectively predicts the step characteristics we can use it for our validation purposes.

A schematic flow of the validation process of the STEPINFO function is described in the figure:

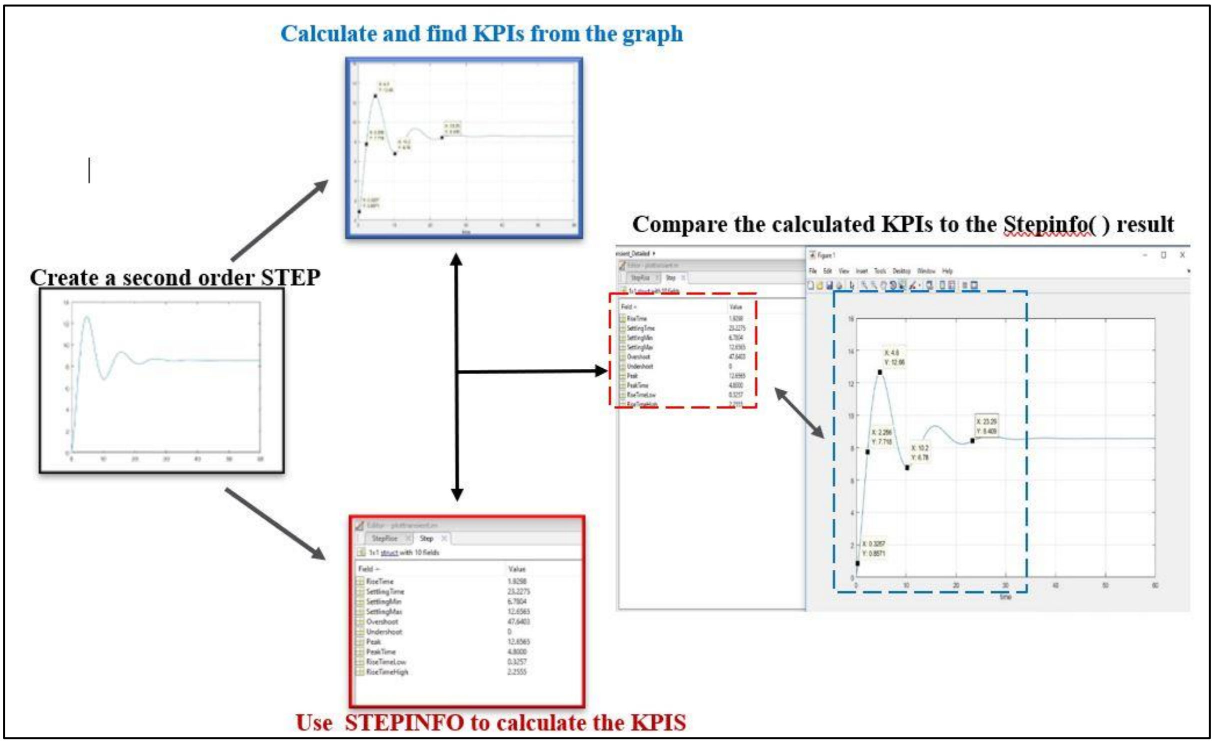


Figure 54 - Stepinfo( ) validation processflow

Now that we understood the process we need to create second order response data which has characteristics similar to the test data. After careful observation of the test data the following type of steps were created:

- Step with a non-zero initial value(We perform the step from a non zero initial condition).
- Step with fluctuations before the start of the step.

Checking the function for a step with non zero initial value:

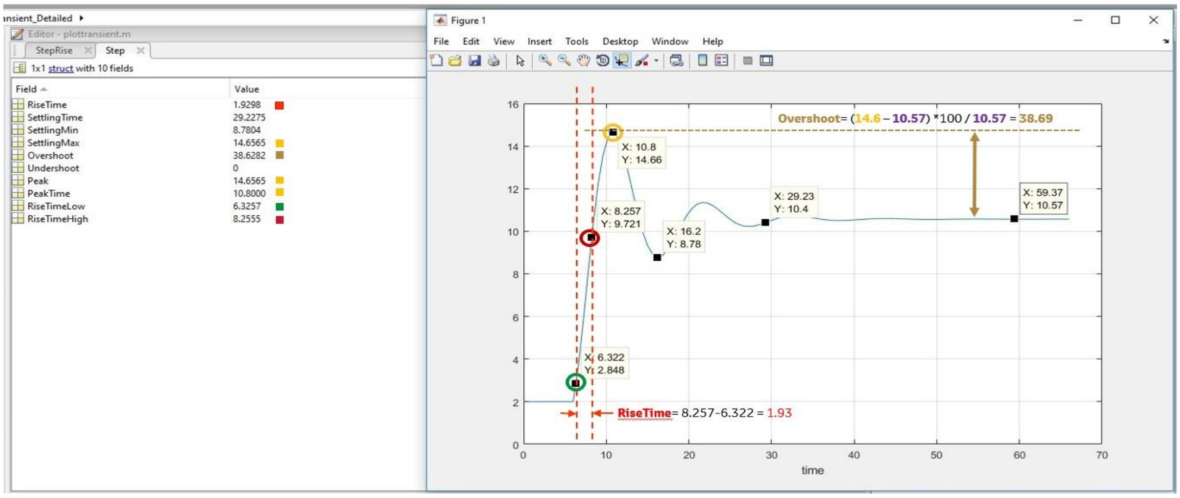


Figure 55 - Step with a non zero initial value

Looking at the above figure and comparing the results it can be seen that the function works good on a step with a non-zero initial value. The color pointers near the values match with the color of the points on the graph. The values can be visually seen from the graph at the x and y co-ordinates. It can be observed that the RiseTime , PeakTime ,Peakvalue , Overshoot etc are correctly predicted as seen on the graph.

Checking the function for a step with a fluctuating behavior before the start of the step:

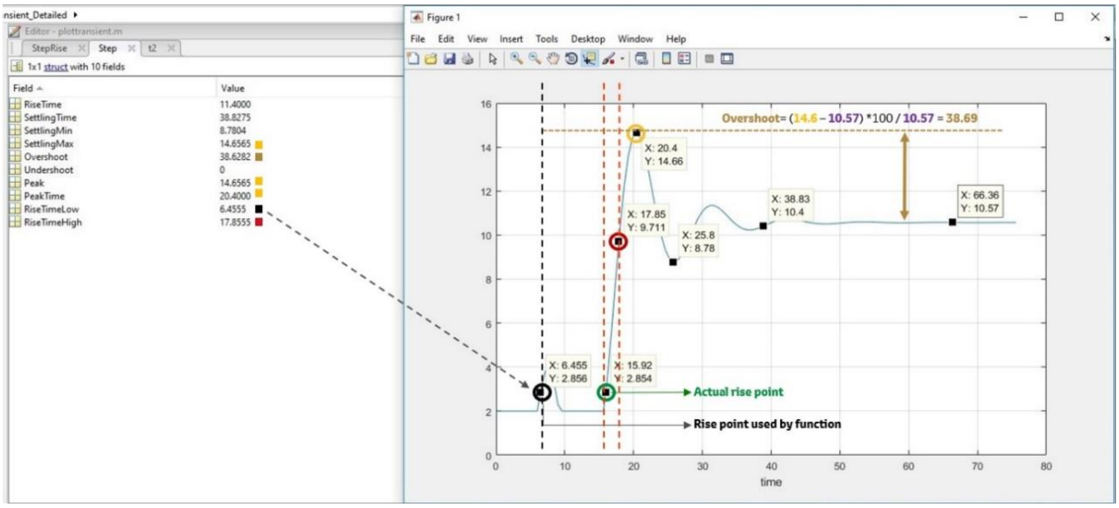


Figure 56 – stepinfo( ) on step with fluctuating behaviour at the start

Here it’s observable that the start of the step or RiseTimeLow is calculated at the black point which is a small fluctuation in the data during the initial phase. This is because the function is designed to calculate at the point where the data starts to rise. But for the validation purposes it should be calculated at the green point where the actual Rise of the step starts. This is one of the limitations of the STEPINFO function. To eliminate this we can crop off the data where the fluctuation occurs and check for the value to the right side of it as illustrated in the figure below.

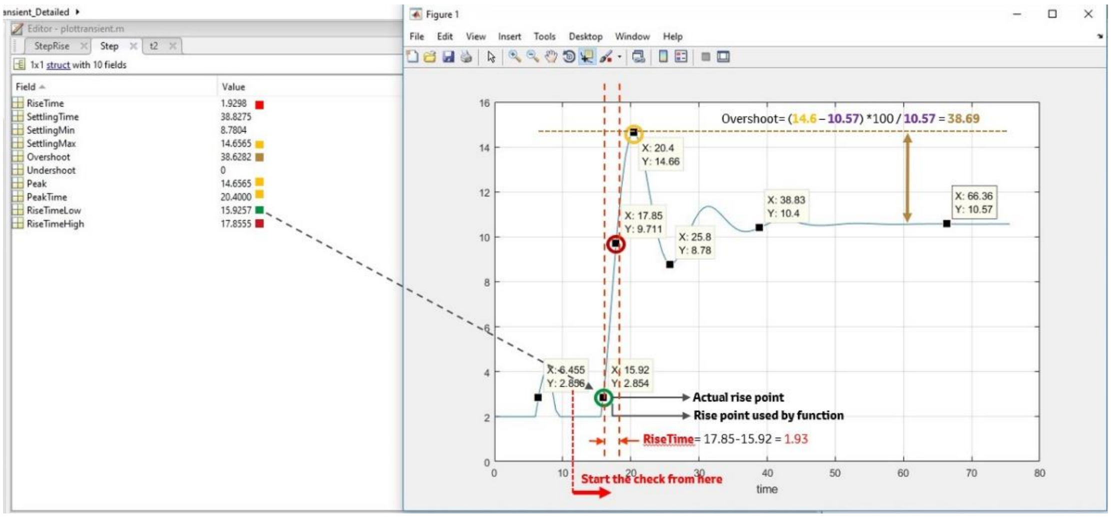
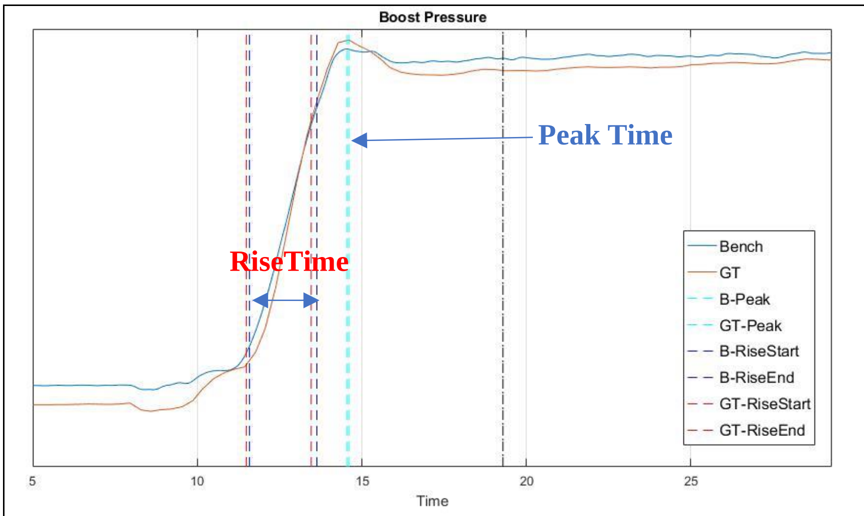


Figure 57 - Calculation by stepinfo( ) after improvement

It is observed in the figure the issue has been fixed now as the function calculates the characteristics to the right side on the red arrow. This fix has to be done manually if the data has such behavior. It can be now safely said that the function can be used to characterize a step response data.

We can apply the above function on the transient data we have and obtain the plot shown in the figure for every test case. The guides show the start and end of the risetime calculation for the bench data and the GT model. It also shows where the peak occurs. The RiseTime parameters are obtained as shown in the figure 55.



Legend description:
Bench: Test data
GT: Simulation data
B-Peak: Peak occurrence of Test data
GT-Peak: Peak occurrence of simulation data
B-RiseStart : Start of Rise of Test data
GT-RiseStart: Start of Rise of simulation data.
B- RiseEnd: End of rise of Test data
GT- RiseEnd: End of Rise of simulation data.

Figure 58 - RiseTime and PeakTime calculated for Test and simulation Data

9.2.2. Determining the KPIs:

Now we know the characteristics of a step response the right KPI for the validation should be identified so the model behavior can be effectively compared with the engine behavior. To do this the transient data were observed from the graphs.

Plotting the results for all the data and analyzing the risetime and peaktime to identify the right KPI the following issue was noted:

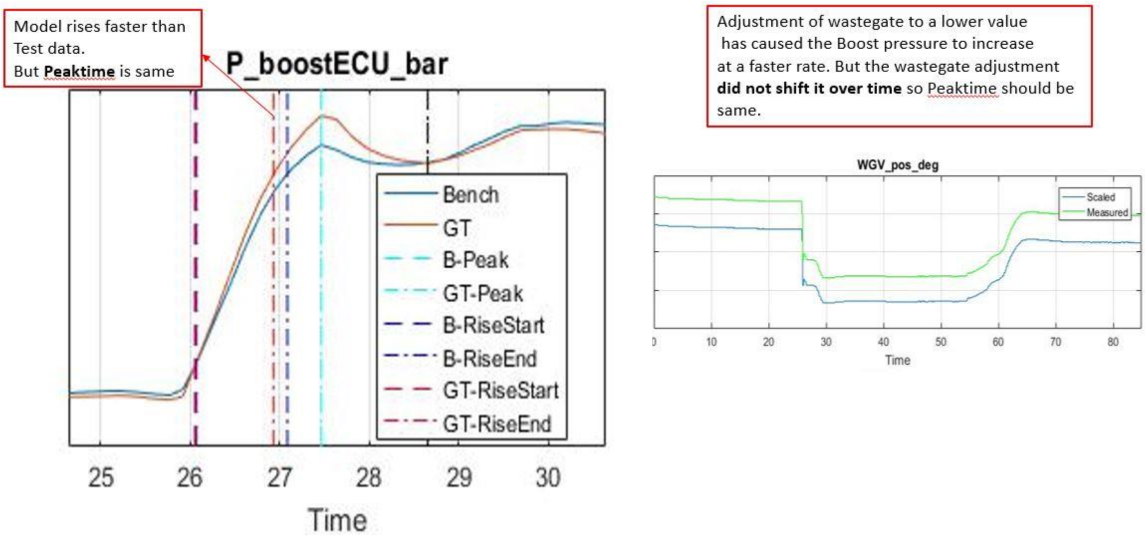


Figure 59 - Issue with RiseTime

Issue:

The issue observed above is that the risetime of the model is different as it can be observed from the boost pressure data on the figure 58. But also it can be seen that the peaks occurs at the exact same time between the test and the simulation data. This poses a question of considering which of the two KPIs are more reliable for the validation. To understand that we need to figure out the cause of why the model reacts faster.

Cause:

In this case the cause of the faster rise of the model is due to the adjustment process of the wastegate angle which was performed. It can be seen the wastegate angle has been shifted down after the scaling process. It has obviously improved the model result but also since the wastegate is being closed more the sensitivity of the Cd increases and drives the boost pressure faster as the wastegate closes further and further. This causes the model to rise faster.

Finalizing the KPI:

But since the wastegate angle is not shifted along the time axis, the peak of both the data should occur at the exact same time. From the above analysis we can conclude that the PeakTime is the most determining performance index for the Validation Process .

The KPI for validation is :

- Absolute PeakTime Error – Seconds .It is difference in the peaktime between the simulation result and test data.

9.2.3. Analysis and Results:

Now that the methodology of the validation is determined we can use the methodology to calculate the KPIs for the transient tests and simulation data we have. We need to analyze the data from all the step test cases to get a meaningful result out of all the transient tests we performed.

To make a meaningful analysis we need to see on what factor the peak error(KPI) is dependent on or the factor which affects the model . After multiple observations of the error cases from the simulation its seen that the peak error can be affected by the speed of the step response. So the peak error can be plotted against the Risetime of the test data.

To understand what it means , we are comparing the error between the peaktime of the model and the engine to the time the engine takes to rise from initial to the final steady part of the step. So the error varies based on how fast the step occurs. Because we performed three different types of step tests, the way the engine reacts to every test is different. This is explained by how fast the ECU requests the load to the engine when we request a load step. To understand this in detail we have to plot the load request (which is translated into a volumetric efficiency request by the ECU) and the actual load(volumetric efficiency) of the engine. It can be seen from the graphs shown below for the request of the load and the actual load for the three steps:

Pedal Step:

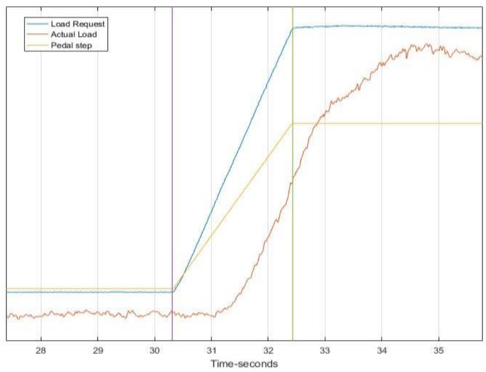
	Delayed load request by more than 1 second as observed. The actual load rises slower.

Figure 60- Pedal Step- Requested vs Actual load

It can be seen the actual accelerator pedal request which is given by a step input is tapered by the ECU before being converted into the request for load(Volumetric efficiency). The load request takes about 1-2 seconds to reach the final value. And the actual volumetric efficiency from the engine is slower.

Volumetric efficiency step:

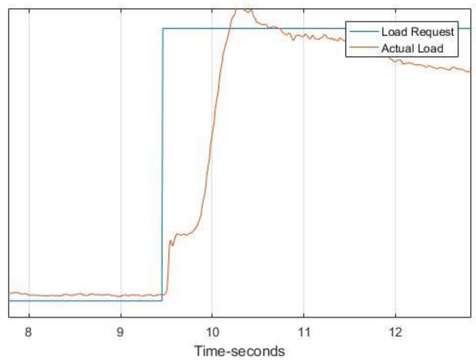
	The request is an instantaneous step signal. The step of the actual load is relatively fast.

Figure 61 - Volumetric efficiency STEP - Requested vs Actual load

As it can be recalled the volumetric efficiency step is performed by giving in a manual request through the ECU. So we basically bypass the ECU to perform the step. It can be seen the volumetric efficiency request is a perfect step signal with no taper. The actual volumetric efficiency rises quite fast in one second.

Boost STEP:

In the boost step the load request is given by modifying the Boost pressure request from the ECU. Based on the boost request the ECU computes the load request.

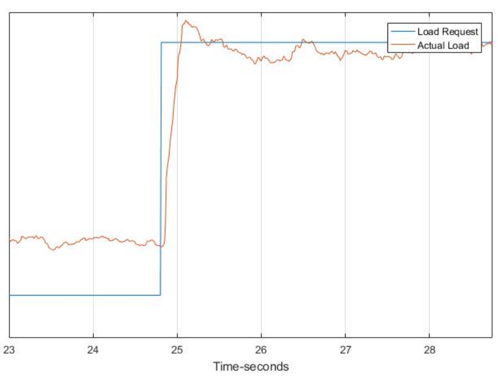
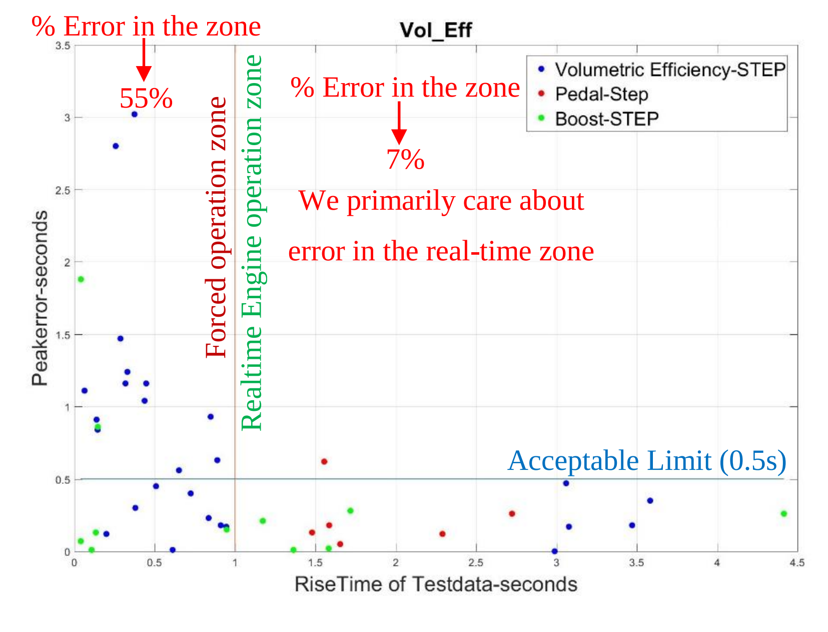
	The request is instantaneous and the actual load rises very fast.

Figure 62 - Boost STEP - Requested vs Actual load

The load rises very fast in the boost step compared to pedal and volumetric efficiency steps. This is only for the certain cases.

It is to be noted not all the volumetric efficiency and Boost steps are very fast. There are a few outliner cases where the engine takes time to rise to the requested load. The above explanation is to support the very quick engine behavior seen from the bench test data.

Looking at the above graphs we can now analyze the model how good it is based on the speed of the step from the engine. So we can plot the error in peak(Simulation-Test data) compared to the risetime of the test as shown.



Title-Volumetric Efficiency

X-axis: The ristime of the parameter from the test in seconds.

Y-axis: The peaktime error between the Model and the test data in seconds

Red Line: The risetime limit to differentiate real time engine operation zone(Right) with the test condition zone.

Blue Line - Acceptable limit on the peak error in seconds.

Figure 63 - Result plot description

Zones: Dividing the graphs into two halves based on the risetime we have two zones of data namely in the Forced operation one and the Realtime Engine operation zone. The zones are divided with a time limit of 1 second. This can be argued from the load request graphs for the three step cases.

RealTime Engine Operation zone: We saw that the pedal step is the slowest with the request delayed by at least 1-2 seconds to reach the target. It can be considered as a real time limit to check the error based on the risetime of the parameters. Importantly the error in this area is of more importance for the validation as this is the area as the MPC is going to operate in this area.

Forced Operation zone: This is the steps done by modifying the load targets through the ECU. The MPC is not going to operate in this area but the error can be calculated and used a secondary KPI. In case the model needs to be improved this error information can be used to predict where the model might behave badly.

Acceptable Limit: The peak time error limit for the model response is set as 0.5seconds. This is decided based on the accuracy needed for control action by the controller. But this accuracy is set arbitrarily initially as we don’t know how much of the model error the controller compensates. This can be explained by the flow in the figure 60. The model is validated based on the initially assumed error limit. If the model is not inside the limit improvements are made to make it better. The controller is developed further based on the model. The controller is tested on the MIL/SIL for performance and validated. If the controller performance is bad then the models need more accuracy. The limit is changed into a more aggressive value and then the models are improved again to satisfy the limit. This loop is performed until the controller achieves good behavior. Normally the controller can compensate for some model errors. But if the models deviate significantly then bad controller behavior can be encountered.

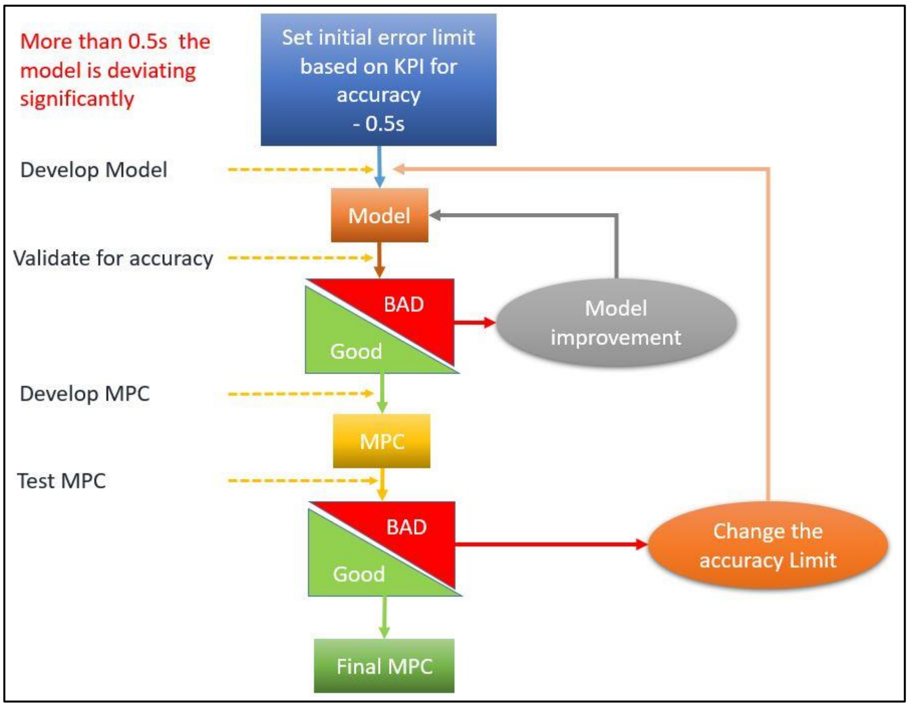


Figure 64 - Error limit for Model accuracy

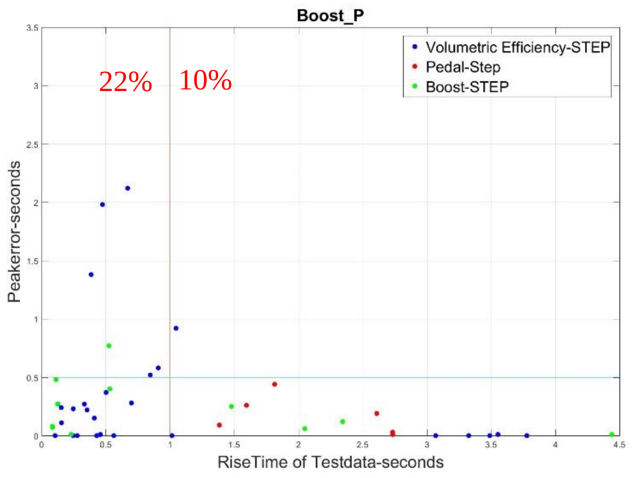
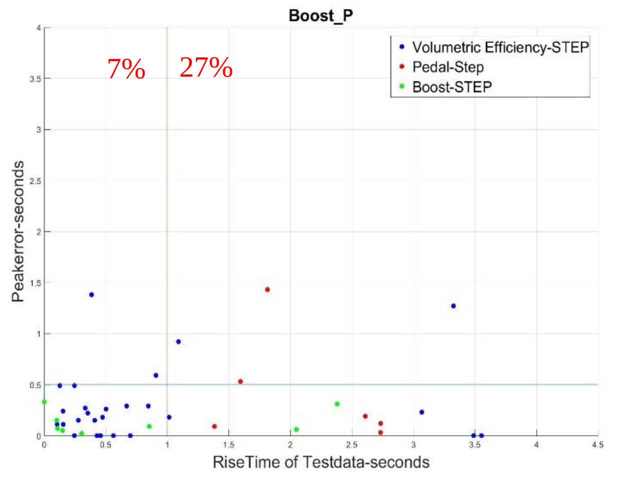
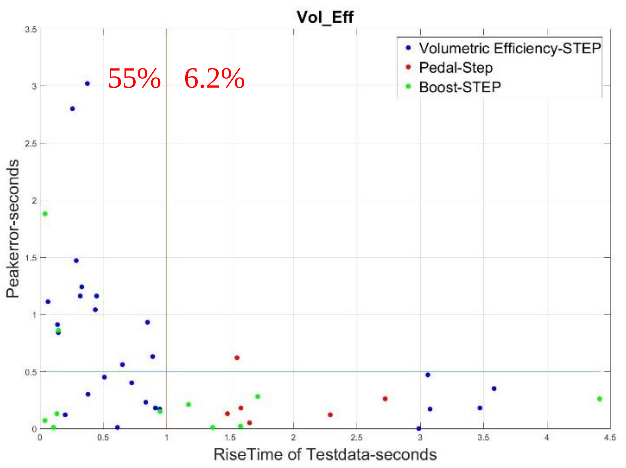
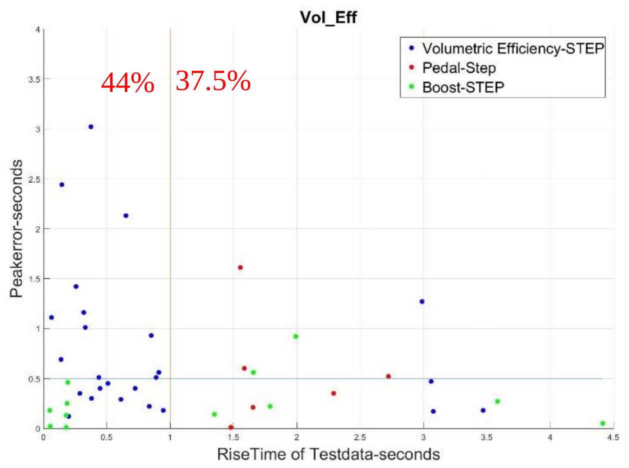
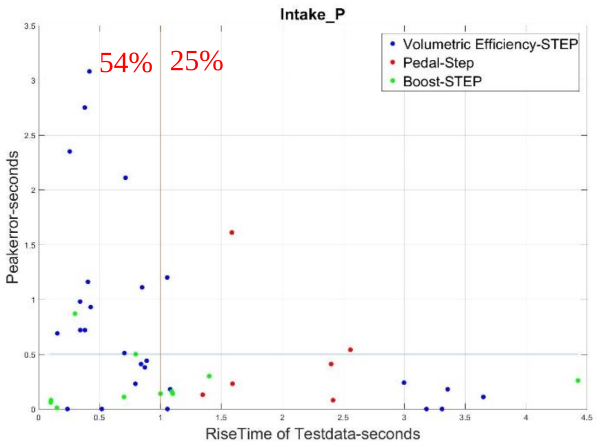
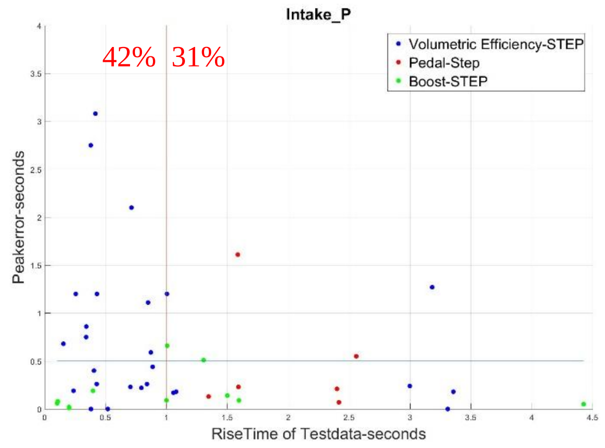
Error : It is the percentage of cases with the peaktime error more than 0.5s . The error percentage is calculated separately for the Real-Time operation zone and the Forced operation zone. But the error which is of primary concern for the validation process is the error in the real-time zone.

The error for the following parameters for all the transient simulation data was considered.

- Boost Pressure-Boost-P(abbreviation used in the plots)
- Intake Pressure-Intake-P
- Volumetric efficiency
- Mass-flow

As the MPC uses the volumetric efficiency and Boost pressure as the setpoints for the controller action it is important to see the prediction error of the model of the two parameters. And the mass-flow and intake are used measured inputs by the MPC for the prediction of the future parameters so it’s important to know how good the model predicts the values.

The results are shown below for the detailed model simulation to the Test data for detailed and MVEM:

Error between Test Data – Detailed Model	Error between Test Data – Mean Value Model
	
	
	

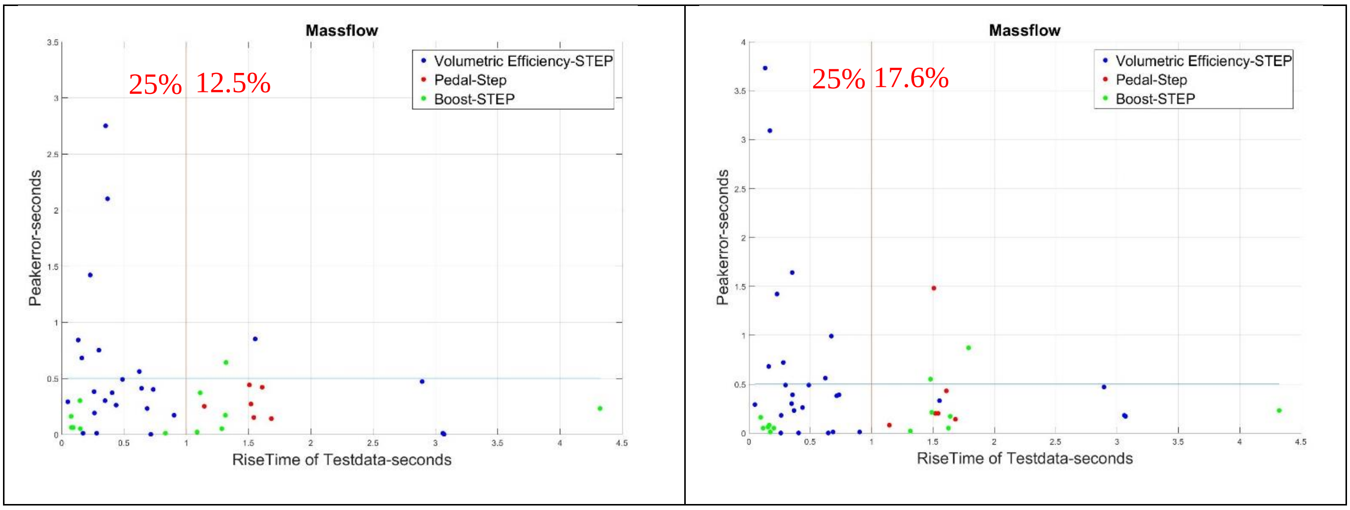


Figure 65 - Transient validation Results

Summary of results:

Table 5- Transient Error Data

	%Error in PeakTime			
Zones	Real-Time Zone		Forced Operation zone	
Models	GT-Detailed	MVEM	GT-Detailed	MVEM
Boost Pressure	10%	27%	22%	7%
Volumetric efficiency	6.20%	37.50%	55.00%	44.00%
Intake Pressure	25%	31%	54%	42%
MassFlow	12.50%	7.60%	25.00%	25.00%

- From the table it can be observed that detailed model has some percentage of error cases in the Real-Time operation zone which is of our primary concern. There is an significant error of 25% cases failing to predict the intake pressure.
- The MVEM has significantly high number of error cases in the Real-Time zone with a maximum of 37.5% cases failing to predict the volumetric efficiency.
- Comparing the models it can be seen the MVEM has higher prediction error in the Realtime operation zone.
- In the forced operation area the GT-Detailed model is significantly bad with high percentage of error in Volumetric efficiency and Massflow .
- The MVEM seems comparatively better but still has higher percentage of error compared to the realtime operation zone.

10.Conclusion:

The objective of the thesis was to identify the model validate the detailed and MVEM and identify the accuracy of the model airpath compared to the engine. This was to make sure the models can be used further for MPC development and MIL testing. From the analysis and results conclusions about the accuracy of the models can be drawn from two simulation and validation processes, namely the steady the transient validation.

From the steady simulation results it was seen the Turbine downstream temperature of the detailed model and the MVEM has a large steady state error. This can be accepted as the parameter on the downstream of the turbine is of no concern to us as they do not affect the model or MPC . All the other output parameters such as the Boost pressure, Intake Pressure ,Volumetric efficiency and the mass-flow have very good correlation. Because from the MPC controller implementation point of view we need the model to predict the right Boost pressure , Volumetric efficiency ,Intake Pressure and mass-flow as the controller uses the volumetric efficiency and Boost pressure as setpoints in the optimisation of the controller output. And the Linear model in MPC uses the Intake pressure and mass-flow as measured inputs to calculate the future outputs and states. The models can be concluded to be accurate at predicting the steady state conditions effectively.

Looking at the transient step response validation results it can be observed that the detailed model has some error in the real-time operating zone when the error limit is set as 0.5seconds. As this is the area of primary interest as the engine operates in real time conditions in this area the model has to be improved to perform completely within the limit. Looking at the MVEM results, it is less accurate than the detailed model. This calls for more improvement as the MVEM is used to create the linear model used in the MPC. In the forced operation zone both the models show a large percentage of error. This can be understood as the models have possibly slower transient because of the turbocharger shaft inertia which is one of the most possible reasons for error in model response.

Further the inaccuracies can be explained by the sensitive Cd-WGV angle model can affect the behaviour at the peak of the step as we did only a linear scaling adjustment to make the model reach the initial and final steady state values. But the transient phase will still have fluctuations in the peak value which can affect the model. In the MVEM the lumping of volumes and a the cylinder model could cause inaccuracies.

Improvements can be made to calibrate the turbocharger shaft inertia to improve the model response. The cd-WGV angle model has to be improved with a new map created based on more accurate measurement data. The lumped volumes in the MVEM should be calibrated.

With the improvements mentioned made the models can be used for further creation of the MPC controller and they can be used in MIL and SIL environments for testing the controller.

11. Appendix

11.1. Appendix A - Generation of Maps over the Operating Range:

The steady testing points are used of the generation of maps of parameters over the operating range. The maps are a 3-D surface plot generated by interpolation and regularizing the values of the operating parameters over the engine speed and volumetric efficiency. The process of map generation is explained below:

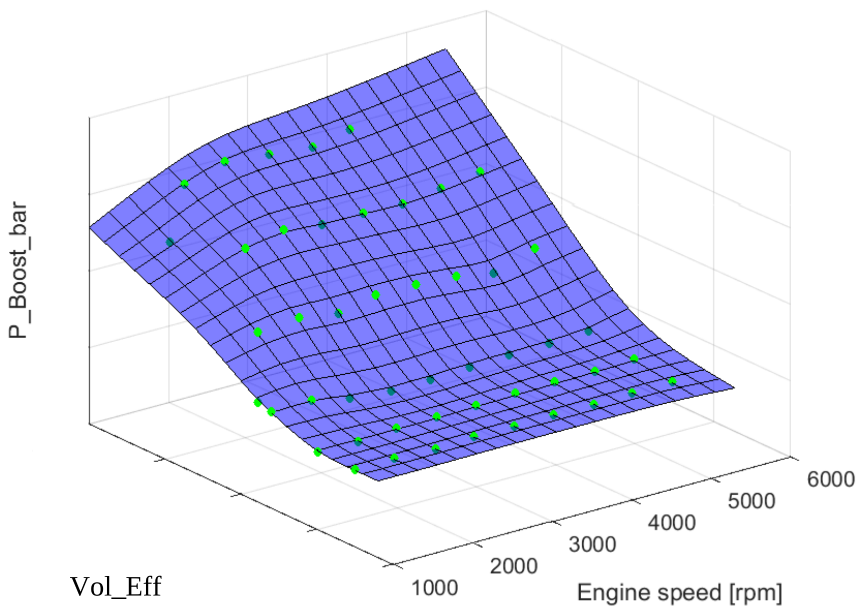
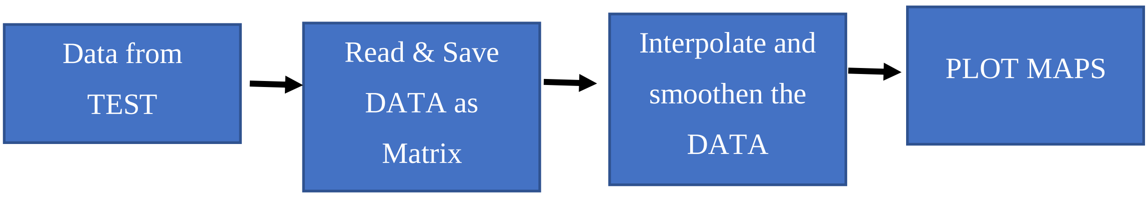


Figure 66- Mapping of Engine Parameters



Raw DATA: It’s the data obtained from the bench logs and ECU logs. It’s time dependent data logged over the period until the engine reaches steady state for every point.

Read and Save DATA: Pick the final steady value for every point for all the engine parameters. Save the data in a structure with parameter fields containing the steady value.

Interpolate and smoothen the DATA: Create a 2D matrix of the parameters over the Engine speed and volumetric efficiency in Matlab. Use a suitable range for the Engine speed and the Volumetric efficiency. Smoothen the data as the original recorded points are discreet.

Generate Maps: Use a surface plot to plot the Maps over the Engine speed and Vol Eff.

11.2. Appendix B : Model in Loop Setup – For Future MPC testing

Model in the loop or MIL testing is a testing method where the test rig comprises of the model and the operating system logic model. In our case the model predictive controller is required to be tested in the MIL environment for initial performance evaluation. The model in loop is a Simulink toolchain consists of the whole vehicle working logic based on sub models from each and every component such as the Engine, ECU , Clutch etc. The toolchain is a virtual test rig for any controller or new logic testing on the engine. The objective of the toolchain in this project is to be accurate so that the developed MPC logic can be tested on the Engine and it's behavior can be analyzed before it can be tested on the actual engine.

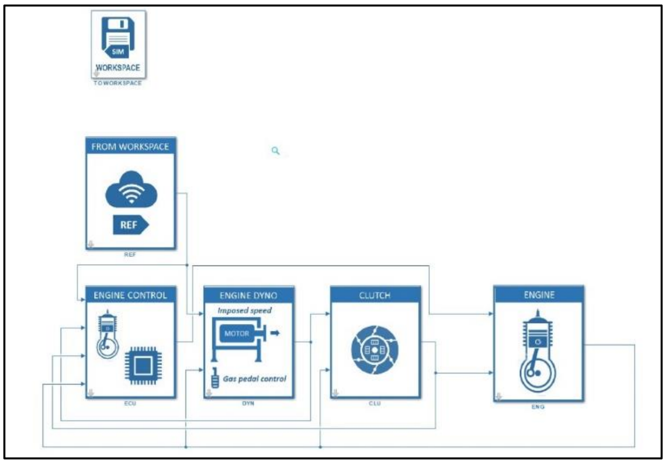


Figure 67 - Toolchain for MIL

The Toolchain used in the setup has a conventional ECU logic in Simulink. This is the logic used in the production engine which is currently used for testing. The main objective of the setup is integrating the Engine model into the Simulink Toolchain. As the engine model we are using is modelled in GT it is required to setup a co-simulation to run the engine model coupled in the toolchain with the conventional logic.

Coupling the GT-Model in the Simulink is possible using a S-function block which can accommodate the GT model and create an interface between the GT and the Simulink. This S-function block is a part which consist of one input port and one output port. The properties of the S-function block consists of the number inputs from the Simulink and out of the model into the Simulink.

The signals into the GT-Model S-function block should be annexed using a MUX block parameter. The signal from the MUX is given as the input into the GT-model. The single output signal from the GT is extracted into the signals we need to acquire using a DEMUX block parameter. The setup of the Engine sub model in the GT-suite is shown below in the figure.

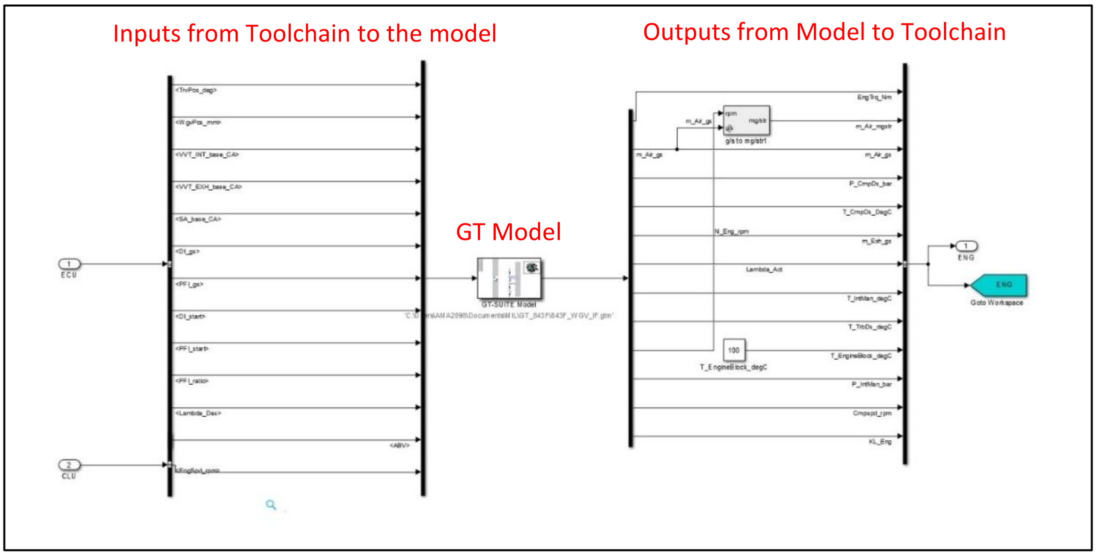


Figure 68- Coupling GT-Model in Simulink

The signals from the Simulink into the S-function GT block is acquired by the GT model using a Simulink harness block in GT-suite. The Simulink harness is similar to a send and receive block in the GT-suite. It is used to interface the Simulink and the model by receiving the required signals into the model and by sending the necessary signal from the model back into Simulink.

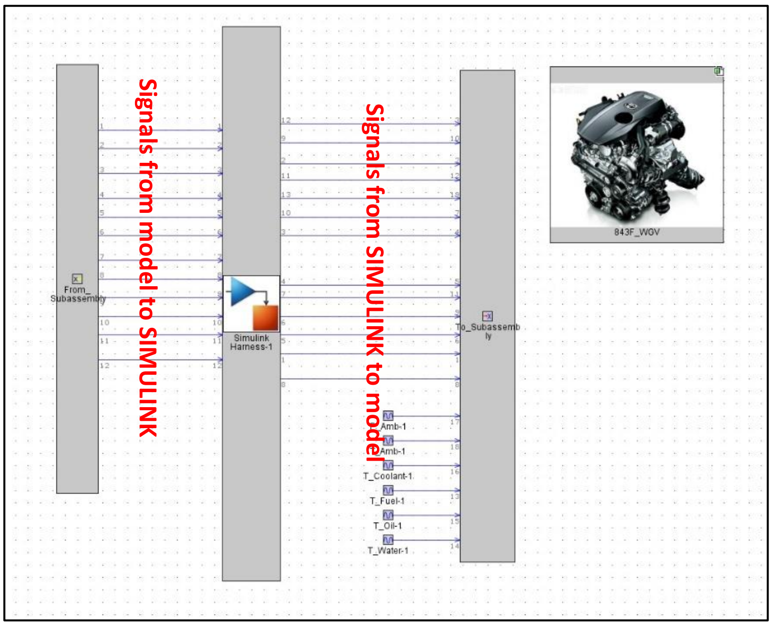


Figure 69 - Setup for Simulink Harness

In the above figure the setup of the simulink harness is shown to interface the simulink and the GT. In the options in the simulink harness it should be chose for the model to run from simulink. Which means simulink is the parent control which send input into the model and the model then calculates using the inputs and sends back the outputs into the simulink. The harness has options for the inputs and outputs. The outputs is the parameters which needed to be sent from the simulink into the model. The order of the outputs should be in the same order the signals are sent into the MUX block parameter which connects to GT S-function block.

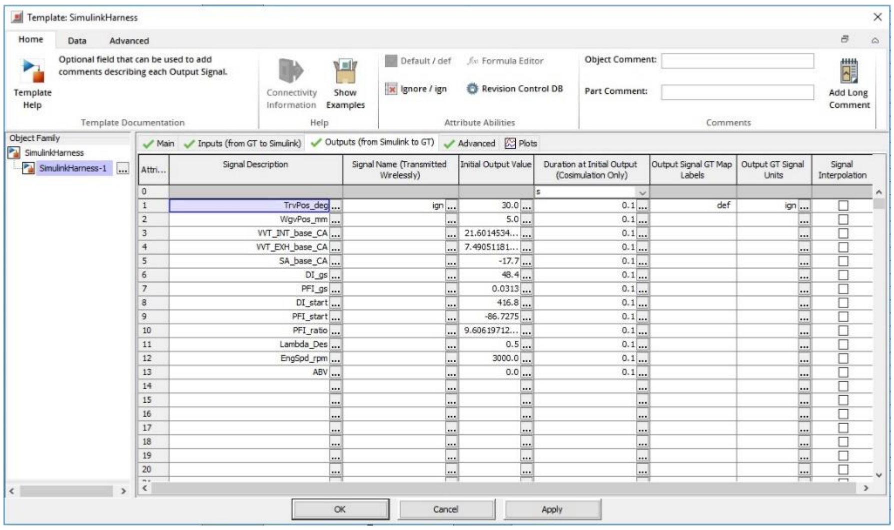


Figure 70 - Input/output setup for Simulink Harness

The timestep is a very important parameter that should be input into the Simulink. A fixed time step solver is chose and both the GT and the Simulink are synchronized.

Some modifications on the existing ECU logic such as a new logic to stabilize the model behavior during the idle conditions was implemented. This is because the model was unstable in the idle conditions due to the turbocharger. The compressor is bypassed during the idle operating conditions to stabilize the model.

MIL setup test results on WLTC drivecycle:

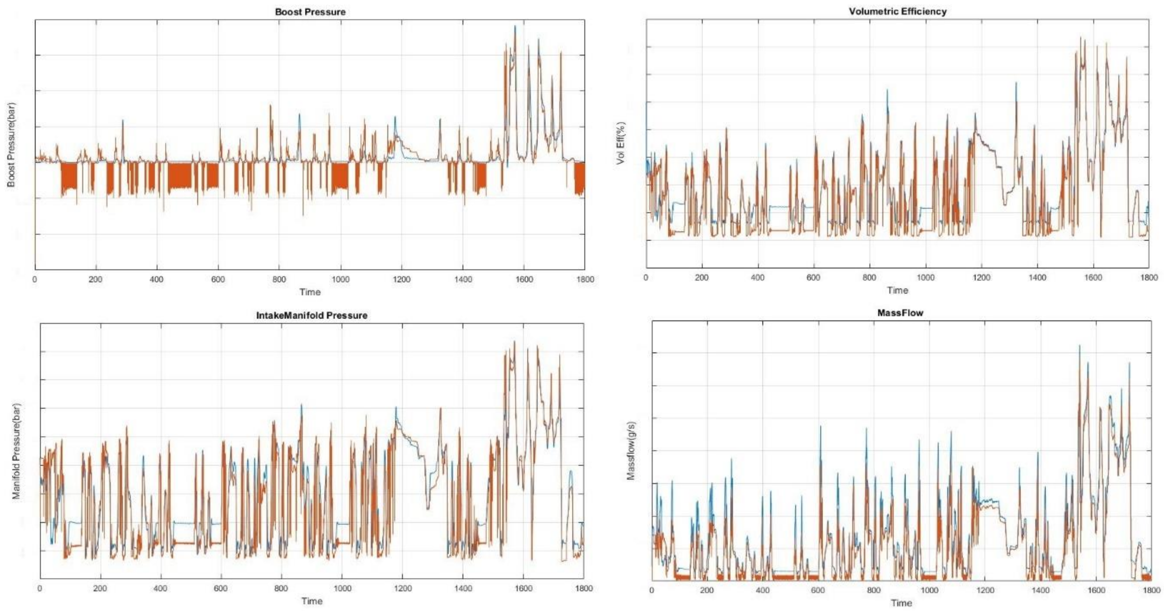


Figure 71 - Toolchain Tested on WLTC cycle

References

1. Automotive Control Systems For Engine, Driveline, and Vehicle, Kiencke, Uwe und Lars Nielsen , Published by Berlin ; Heidelberg ; New York : Springer (2005)
2. MPC: Current Practice and Challenges ,Mark L.Darby, MichaelHarmse, MichaelNikolaou
3. Ali, E., and Zafiriou, E., “Optimization-based Tuning of Non-linear Model Predictive Control with State Estimation”, J. of Process Control, 3, 97-107, 1993.
4. Lipták, Béla G. (2003). Instrument Engineers' Handbook: Process control and optimization (4th ed.). CRC Press. p. 108. ISBN 0-8493-1081-4.
5. Kevin Forsberg and Harold Mooz, "The Relationship of System Engineering to the Project Cycle", in Proceedings of the First Annual Symposium of National Council on System Engineering, October 1991: 57–65.
6. “Computational Fluid Dynamics: The Basics with Applications” by John D Anderson
7. Morel, T., and Keribar, R., "A Model for Predicting Spatially and Time Resolved Convective Heat Transfer in Bowl-in-Piston Combustion Chambers," SAE Paper 850204, 1985.
8. Woschni, G., "A Universally Applicable Equation for the Instantaneous Heat Transfer Coefficient in the Internal Combustion Engine," SAE Transactions, Vol. 76, p. 3065, 1967.
9. Internal Combustion Engine Fundamentals" by John B. Heywood
10. Hires, S.D., Tabaczynski, R.J., and Novak, J.M., "The Prediction of Ignition Delay and Combustion Intervals for a Homogeneous Charge, Spark Ignition Engine," SAE Paper 780232.
11. Blizard, N.C. and Keck, J.C., "Experimental and Theoretical Investigation of Turbulent Burning Model for Internal Combustion Engine," SAE Paper 740191, 1974.
12. Morel, T., Rackmil, C.I., Keribar, R., and Jennings, M.J., "Model for Heat Transfer and Combustion in Spark-Ignited Engine and Its Comparison with Experiments," SAE 880198, 1988.
13. SAE paper 905018 - Investigations on heat transfer in internal combustion engines under low load and motoring conditions. Huber,K. Woschni,G. Zeilinger,K.
14. Chen, S.K., and Flynn, P.F., "Development of a Single Cylinder Compression Ignition Research Engine," SAE Paper 650733
15. Heij, De Boer, Franses, Kloek, and Van Dijk: Econometric Methods with Applications in Business and Economics. (section 3.1)

16. Willmott, Cort J.; Matsuura, Kenji (December 19, 2005). "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance". Climate Research
17. GT-Suite – Engine Performance and Flow manual