TSFS09 2C

November 21, 2025

1 Introduction

This project is similar to 1C. We have two parts:

- Integrating the turbocharger into our engine model.
- Designing a control system, all the way from the gas pedal to the engine torque (more on this later).

2 Integrating the Turbocharger

From project 1, you now have a Simulink model of a naturally aspirated engine. We will now add a turbocharger to it. That is, we will include the parts drawn in red in Figure 1.

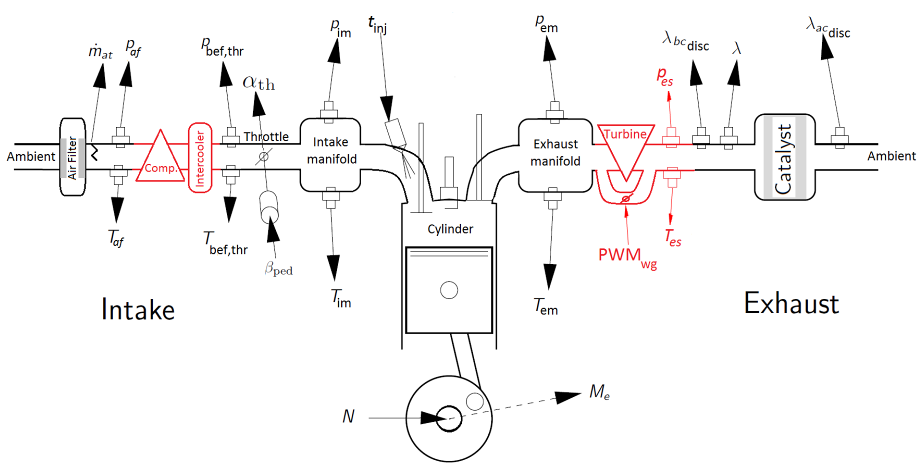

Figure 1: An overview of the air path with signals and components to be modeled in Project 1 (shown in black) and Project 2 (shown in red+black)

2.1 Preparation

- **Before making any changes, save a copy of your naturally aspirated engine model.**

From your work so far, you might've noticed a pattern: we always interleave volume models (intake manifold, exhaust manifold) and flow models (throttle, cylinder flow, catalyst flow). The air path of our engine model forms a chain of manifold → flow → manifold → flow, and so on. To continue this pattern as we introduce the turbocharger models, we must also introduce two additional manifolds:

- an **intercooler volume** between the compressor and throttle,
- and an **exhaust system volume** between the turbine and catalyst.

These subsystem, as well as empty subsystems for the compressor, turbine, and other components, are available in the `turbocharger_models.slx` file.

Before we start to integrate the turbocharger, we must make two adjustments:

- Reduce the displacement volume (V_D) from 2 litres to 1.2 litres. Make sure to adjust every instance of V_D throughout your model (engine torque, cylinder flow, fuel controller feed-forward).
- Reduce the throttle coefficient a_0 to $0.6 \cdot 10^{-5}$.

The first adjustment is made to downsize the engine. A large engine equipped with a turbocharger will act as an even larger engine. A small engine with a turbocharger will feel like a large engine, with improved fuel efficiency.

The second adjustment is made to reduce the idling speed of the engine.

With these adjustments, without a turbocharger, the downsized engine is too weak to follow the drive cycle, as shown in Figure 2.

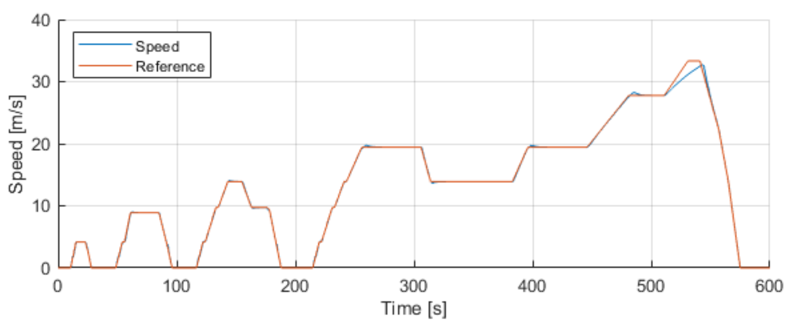

Figure 2: A downsized engine has trouble following the velocity profile.

2.2 Compressor Flow

We start by introducing the compressor flow model.

- Find the `compressor` and `intercooler volume` subsystems, deduce the correct Simulink signal connections according to Figure 1, and connect it to your engine model.
- Complete the compressor flow model (ignore the output `W_dot_c` for now). Assume $T_{amb} = T_{01} = 273.15 + 25$.

Since we have no turbine, nothing drives the compressor shaft. However, we can pretend for a moment that a very strong electric motor powers the compressor, and is controllable such that we can request any rotational speed.

- For ω_{tc}, begin with a small value of 1000 rad/sec. Increase the value until the vehicle is able to follow the drive cycle.

At low speed, the compressor supplies insufficient air for the engine to follow the drive cycle, as shown in Figure 3. At $\omega_{tc} = 8000$ rad/sec, shown in Figure 4, the engine should output enough torque.

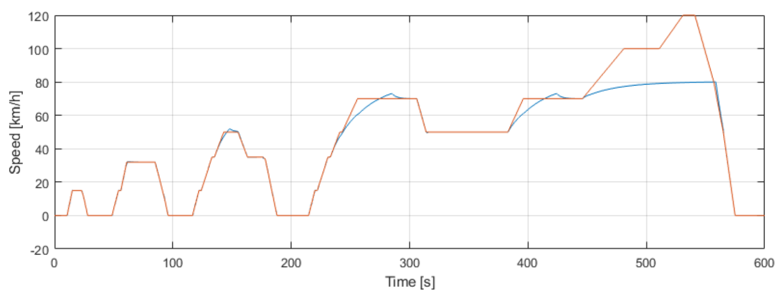

Figure 3: Velocity tracking when $\omega_{tc} = 1000$ rad/sec.

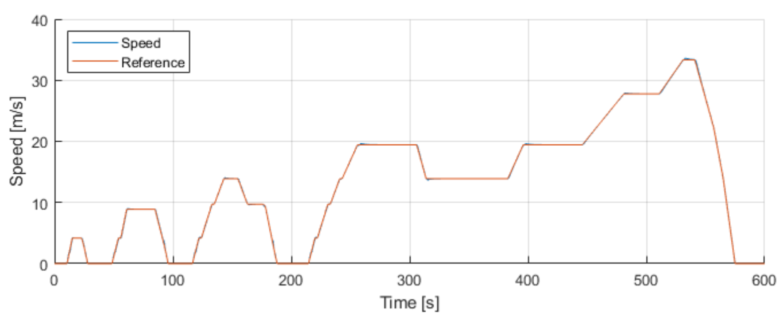

Figure 4: Velocity tracking when $\omega_{tc} = 8000$ rad/sec.

2.3 Compressor Efficiency

Finish the compressor subsystem:

- Introduce the compressor efficiency, temperature, and power models. Add a saturation on the compressor efficiency, clamping the signal between 0.3 and 1.

The compressor power $\dot{W}_c$ is shown in Figure 5, for $\omega_{tc} = 8000$ rad/sec.

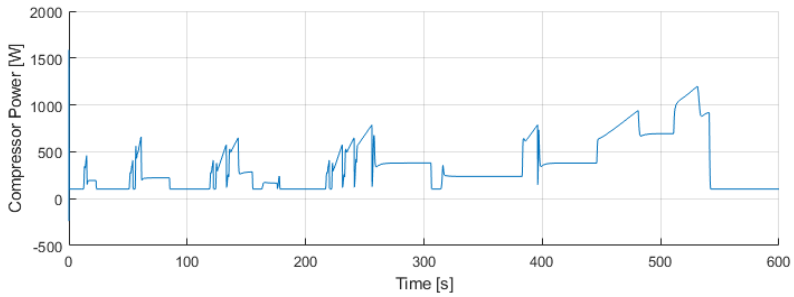

Figure 5: Compressor power ($\dot{W}_c$) when $\omega_{tc} = 8000$ rad/sec.

2.4 Turbine Flow

With the compressor finished, we will turn to the turbine.

- From the `turbocharger_models.slx` file, grab the `turbine`, `wastegate`, and `exhaust system volume` subsystems. Connect the signals according to Figure 1. Assume a fully open wastegate, i.e. a constant $u_{wg} = 1$, and $\omega_{tc} = 8000$ rad/sec. For the catalyst flow model, assume an inlet temperature of 700.
- Model the turbine flow in the `turbine` subsystem. Ignore `W_dot_t` for now.

In Figure 6, the exhaust manifold pressure (p_{em}) is shown for three constant values of u_{wg}. Try these values and compare your results. Can you explain why the pressure is significantly higher for $u_{wg} = 0$?

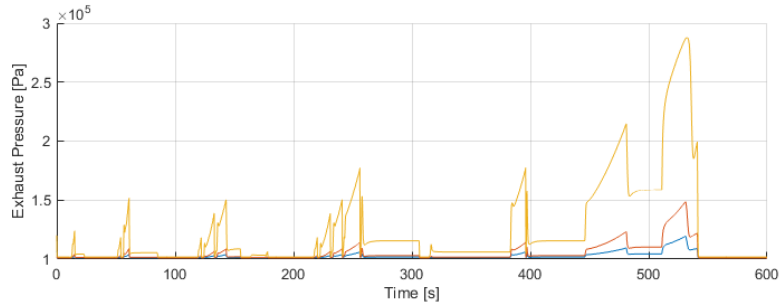

Figure 6: Exhaust manifold pressure (p_{em}) when $\omega_{tc} = 8000$ rad/sec, for three values of u_{wg}.

2.5 Turbine Efficiency

Finish the turbine subsystem:

- Introduce the turbine efficency, temperature, and power models. Add a saturation on the turbine efficiency, clamping the signal between 0.3 and 1. Connect the turbine outlet temperature to T_{es}.

2.6 Shaft Dynamics

Now it is time to connect the turbine and compressor.

- Introduce the shaft dynamics subsystem. Connect the Simulink signals and finish the shaft dynamics model. Replace the constant turbocharger speed blocks with the output of the shaft dynamics subsystem.

Running the simulation produces the intake and intercooler pressure signals shown in Figure 7 and 8.

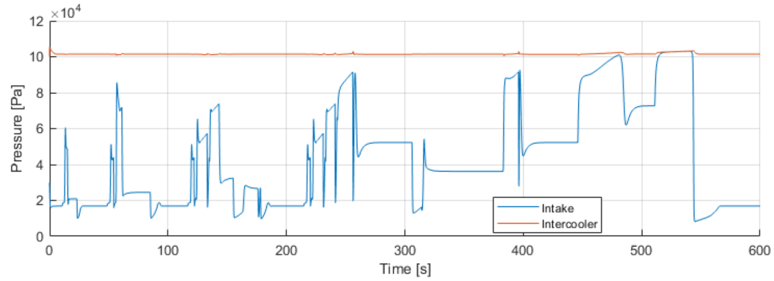

Figure 7: Intake manifold and intercooler pressures for $u_{wg} = 1$

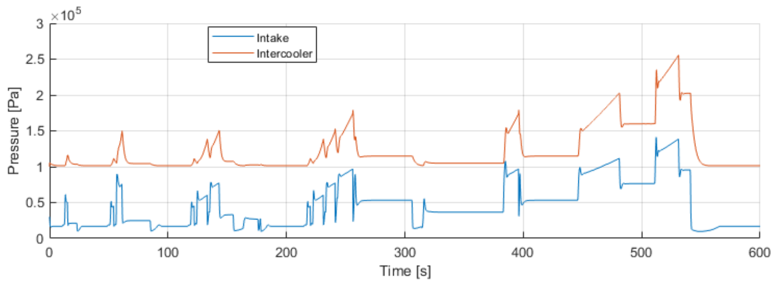

Figure 8: Intake manifold and intercooler pressures for $u_{wg} = 0$

3 Driver Gas Pedal Interpretation

With the turbocharged engine model complete, we can now move on to control.

First:

- Turn on boost control by setting the variable `Boost_control` = 1 in the model initialization script.
- In the boost control subsystem (ECU), turn off the throttle and wastegate feedback through the manual switches.
- Connect the `wg_REF` input to the wastegate in your model.

For naturally aspirated engines, the gas pedal can be linked directly to the throttle (as it was, mechanically, for older engines). More gas pedal actuation → more air to the engine → more torque on the wheels.

For turbocharged engines, this is no longer true. We now have two actuators that both determine the amount of air supplied to the engine: the throttle and the wastegate. Therefore, we cannot directly link the gas pedal to the throttle, as we did in Project 1. We must do something smarter.

3.1 Pedal Actuation → Torque Request

The trick is to interpret what the driver wants: a small gas pedal actuation should provide an equally small response in torque. A large gas pedal actuation should provide an equally large response in torque.

Consequently, we need to map the *gas pedal actuation* to a requested *engine torque*. This we do through look-up maps, shown in Figure 9.

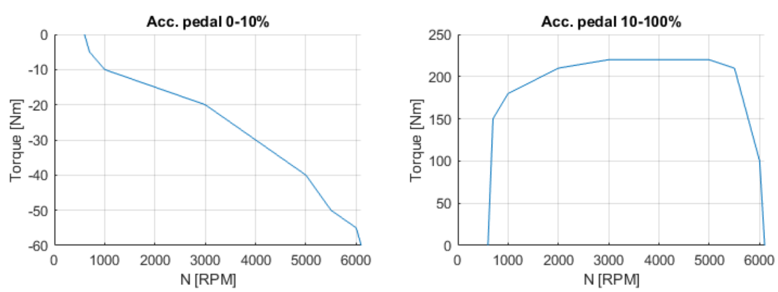

Figure 9: The maximum available torque output for varying engine speeds.

Suppose that the engine speed is $N = 2000$ RPM. If the driver actuates the gas pedal to 60 %, we may interpret that as a request for roughly 66 % of the available maximum torque when $N = 2000$. (It will be $60\% \times 1/0.9 = 66.666\ldots\%$).

Instead, if the gas pedal is actuated to 5 %, we may interpret that as the driver requesting negative torque (engine braking).

- Use the proposed diagram below (Figure 10) to implement your gas pedal interpretation. The torque-speed maps are found in the file `TqEvsNeMAP.mat`. Use `1-D Lookup Table` and `switch`-blocks. Set the threshold-parameter in the switch to 0.1.

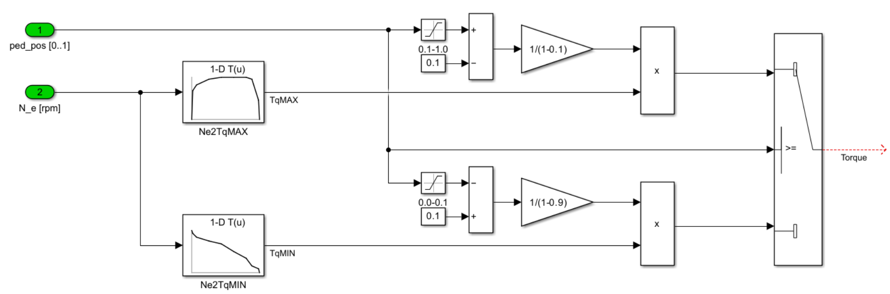

Figure 10: A proposed Simulink diagram for the gas pedal interpretation.

3.2 Torque Request → Intake Pressure Reference

We could start tuning our regulators now, but torque is a difficult quantity to measure and control directly, and therefore, we would much rather work with something else, namely the *intake manifold pressure*.

The air pressure in the volume just before the cylinder is closely related to the engine output torque, and thus can be correlated by a simple linear model. However, since our data is recorded on a 2.0 litre engine, we instead use the brake mean effective pressure (BMEP) which is engine-size independent.

Use the following model for BMEP:

$$\text{BMEP} = C_1 \cdot p_{im} - C_0, \qquad\qquad \text{BMEP} = \frac{2\pi n_r}{V_D} \cdot M_e$$

where $C_0 = 186920$, $C_1 = 11.8212$.

- Implement the BMEP model in Simulink to calculate the reference intake pressure $p_{im,ref}$ from the requested engine torque (continuing from your torque interpretation).

3.3 Intake Pressure → Boost Pressure Reference

So far we have not dealt with the fact that we have two degrees of freedom (two actuators that control the same output): we need to add a second control objective.

One of the drawbacks of turbocharged engines is the lag: it takes time for the shaft to spin, so the torque response is noticeably slower than larger, naturally aspirated engines. One method to mitigate the lag-effect is to constantly maintain a higher turbocharger speed than necessary, building up a "boost pressure". Then, when the driver suddenly requests a large torque, there is a storage of high-pressure air ready to supply the new demand.

On the other hand, this strategy will result in larger losses, and therefore a poor fuel efficiency. In fact, an optimal strategy for fuel efficiency is to use the turbocharger as little as possible.

To produce a happy medium, we can blend these strategies by controlling the pressure difference between the intake manifold and intercooler (boost) pressure. With our $p_{im,ref}$ computed, we can simply add a desired constant to set our boost pressure reference $p_{ic,ref}$, i.e.,

$$p_{ic,ref} = p_{im,ref} + \Delta p.$$

- Finish the driver gas pedal interpretation subsystem by choosing a constant $\Delta p = 10$ kPa.

With two control objectives (the intake and intercooler pressures) we may begin with our controllers.

4 Boost Control

The control objective now is to track the desired intake and intercooler pressure references.

In the `ECU/Boost Control` environment, there are three subsystems of note:

- Throttle Feed Forward
- Throttle Feedback including anti-windup
- Waste-gate Feedback including anti-windup

4.1 Throttle Feedforward

Like the fuel controller, we will use both feedforward and feedback for the throttle.

- Complete the `Cylinder air flow REF` subsystem. Use a constant volumetric efficiency $\eta_{vol} = 0.7$ (or the same value as in your fuel controller.)
- Use the `m_dot_ac_REF` signal to compute a corresponding throttle position α_{ref}. For your convenience, the inverted throttle equation is given below.

$$\alpha = \sqrt{\frac{A_{\text{eff}} - a_0}{a_2} + \left(\frac{a_1}{2a_2}\right)^2} - \frac{a_1}{2a_2}$$

Note: Use a saturation on A_{eff}. The effective area cannot be smaller than a_0, and cannot be larger than $a_0 + a_1 + a_2$. Remember to use the new value for $a_0 = 0.6 \cdot 10^{-5}$.

- Run the simulation with only throttle feedforward. Verify that your actual intake manifold pressure and your computed reference are close when the pressure is below 1 atm. Compare your result with Figure 11.

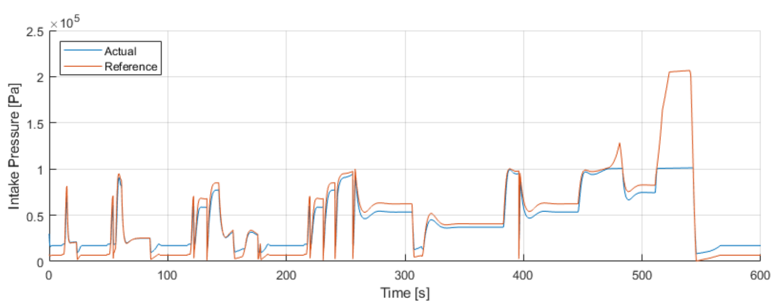

Figure 11: Intake manifold pressure and reference with only feedforward.

4.2 Feedback

The feedback controllers are tuned by two parameters each, `KpThr` and `TiThr` for the throttle, `KpWg` and `TiWg` for the wastegate. The values can be set in the `init_model.m` script.

- Turn on the throttle and wastegate feedback switches.
- Use the values given in the script as a starting point. Simulate while inspecting the reference and actual pressures. Tune the controller parameters until you are satisfied with the tracking performance. Compare your results to Figures 12 and 13.

Note: There is no hard limit on the performance of your controller: try your best to achieve reasonable tracking compared to the figures.

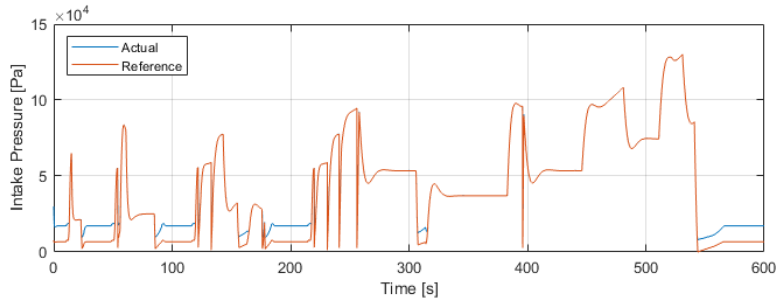

Figure 12: Intake manifold pressure and reference with feedforward and feedback.

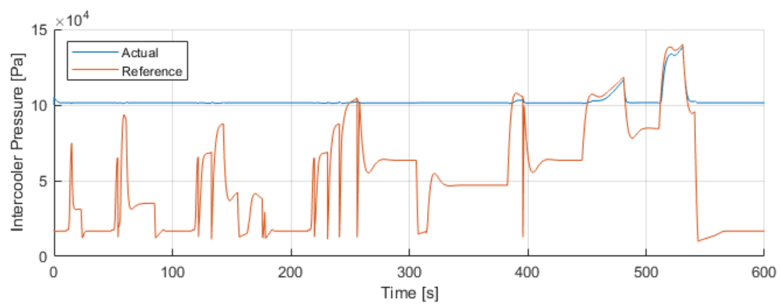

Figure 13: Intercooler pressure and reference with feedforward and feedback.

5 Performance, Emissions & Fuel Consumption

Now, we analyze the engine performance in the same way as project 1.

- Plot the velocity along with the reference speed of the drive cycle. Plot also the intake manifold and exhaust manifold pressure signals. Plot also the engine torque.

In the `emissions`-folder, you will find the function `calcEmissions()`, which you can use to estimate the produced emissions of your engine.

- Calculate the emissions produced by your engine during a 600s drive cycle.

- Finally, compute the fuel consumption of your vehicle in the unit litre gasoline per 100 km. Use $\rho = 780$ [kg/m^3] as density for gasoline.

- Increase the pressure difference over the throttle, Δp, from 10 kPa to 30 kPa. Compute the fuel consumption again.