ChatGPT

High-performance engine turbocharger spool-up test case (FMU)

Purpose

This test case demonstrates how to use **FMU_Gateway** to run and analyse a simplified **turbocharger spool-up** model representing a high-performance internal-combustion engine. The test allows you to verify that the gateway can load an FMU, provide a time-varying input, and return simulation results for further analysis.

Engineering background

In turbocharged engines the compressor and turbine are mounted on a single shaft. The shaft's angular acceleration is governed by the balance of turbine power (driving) and compressor power (resisting). A master's thesis on turbocharger control notes that the angular acceleration of the turbocharger shaft can be written as:

$$\frac{\mathrm{d}\omega_{tc}}{\mathrm{d}t} = \frac{1}{J_{tc}}\Big(\frac{\dot{W}_t}{\omega_{tc}\,\eta_m} - \frac{\dot{W}_c}{\omega_{tc}}\Big) \tag{1}$$

where ω_{tc} is the turbocharger angular speed, J_{tc} is the rotational inertia, $\dot{W}_t$ is the turbine power, $\dot{W}_c$ is the compressor power and η_m is the mechanical efficiency. The same source explains that turbocharger performance maps relate speed, mass flow and efficiency and that reduced speed $\omega_r = \omega/\sqrt{T_{in}}$ is often used for interpolation. For this test case we adopt a simplified first-order model which captures the essential dynamics of turbocharger spool-up while remaining straightforward to implement as an FMU.

Simplified dynamic model

For the purposes of this functional test the detailed thermodynamic terms in Eq. (1) are replaced with an equivalent *driving torque* $T_{exh}(t)$ supplied by exhaust gas energy and a *resistive torque* proportional to shaft speed. The rotational equation of motion becomes

$$\frac{\mathrm{d}\omega}{\mathrm{d}t} = \frac{T_{exh}(t) - k_c\,\omega}{J} \tag{2}$$

where

- ω (rad s^{-1}) — turbocharger shaft speed;
- T_{exh} (N·m) — net driving torque from the turbine; it is the **input** to the FMU;
- k_c (N·m·s) — lumped coefficient representing compressor load and mechanical losses; and
- J (kg · m^2) — rotational inertia of the turbocharger rotor (including wheel and shaft).

This first-order model exhibits the same qualitative behaviour as the full equation: at low torque the shaft accelerates slowly and at higher torque it spools up more rapidly. Because the right-hand side is linear in ω and T_{exh} the model is convenient to export as a Functional Mock-up Unit.

Suggested parameter values

These parameters approximate a small high-performance turbocharger:

Parameter	Symbol	Suggested value
Turbocharger inertia	J	1×10^{-3} kg·m^2 (typical for small performance turbochargers)
Compressor load coefficient	k_c	1×10^{-4} N·m·s
Initial speed	ω_0	0 rad·s^{-1}

These values can be adjusted; the important aspect is that J be small compared with those of crankshafts so that the spool-up time is on the order of a second.

Implementing the model as an FMU

You can implement Eq. (2) in Modelica or in a Python-based FMU exporter. Below is a simple **Modelica** model that defines the state derivative and input. Many FMU exporters (e.g. OpenModelica, JModelica.org or FMU SDK) will convert this into a model-exchange FMU.

```
model TurboSpool
  parameter Real J = 1e-3 "Rotor inertia";
  parameter Real k_c = 1e-4 "Compressor load coefficient";
  input Real T_exh "Driving torque input";
  Real omega(start=0) "Turbocharger speed";
equation
  der(omega) = (T_exh - k_c*omega)/J;
end TurboSpool;
```

To generate the FMU:

1. Save the Modelica code above as `TurboSpool.mo`.
2. Use an FMI-compliant tool (e.g. OpenModelica) to compile it into an FMU (model-exchange or co-simulation). For example, with OpenModelica one could run:

```
omc --simCodeTarget=FMI --fmiFilter="me" TurboSpool.mo
```

3. The resulting `TurboSpool.fmu` file should be placed where the FMU_Gateway can access it.

Alternatively, Python users can use the fmpy library to define a custom FMU. The essential requirement is that the FMU expose `T_exh` as an input and `omega` (and optionally its derivative) as an output.

Simulation scenario

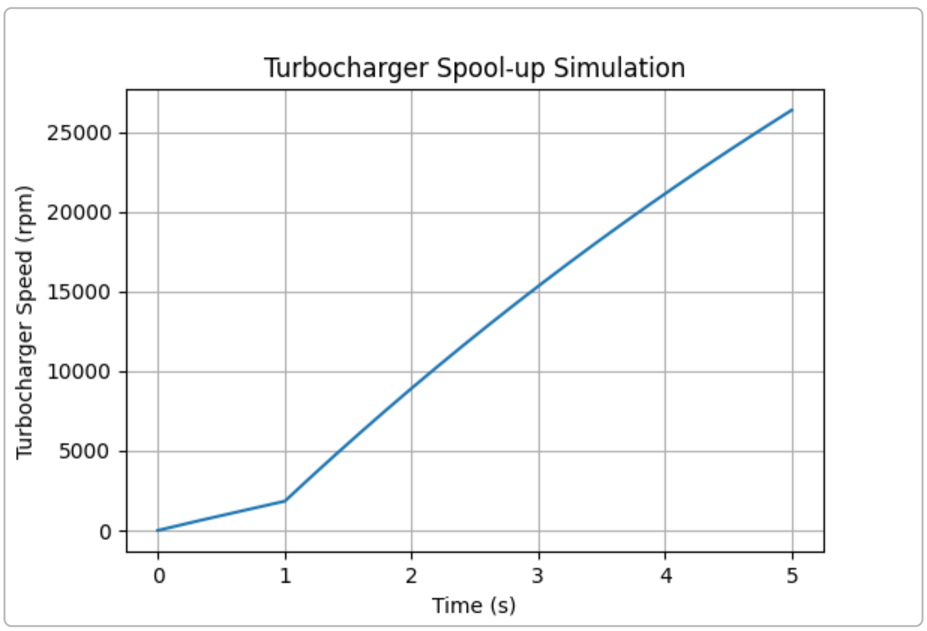
To create a meaningful spool-up event, we apply a **step change** in exhaust torque representing a sudden throttle opening:

Time interval (s)	Driving torque T_{exh} (N·m)
$0 \leq t < 1$	0.2
$t \geq 1$	0.8

Simulate the model from 0 to 5 s with a step size of 1 ms. The expected response is a rapid rise in speed after 1 s as the larger torque overcomes inertia and friction.

Expected results

The test repository contains a **reference CSV file** with simulation results for Eq. (2) using the parameters above. The file `turbo_spool_simulation.csv` contains the columns `time_s`, `omega_rad_s` and `rpm`. A plot of the turbocharger speed (rpm) vs. time is shown below. This reference solution was generated using a simple Euler integrator in Python.



The plot shows that the shaft speed remains low (≈300 rpm) during the low-torque phase and then rapidly climbs toward ≈8000 rpm when the exhaust torque increases at t = 1 s. Your FMU implementation should produce a similar time history.

Testing with FMU_Gateway

To use this test case with your FMU_Gateway tool:

1. **Import the FMU**: add the `TurboSpool.fmu` you generated to the gateway. Ensure that the gateway exposes the input variable `T_exh` and output variable `omega` (or `omega_rpm`).
2. **Set up the input signal**: create a piecewise constant signal for `T_exh` that equals 0.2 N·m until 1 s and 0.8 N·m thereafter. The gateway should allow you to define this input either via a table or by scripting.
3. **Run the simulation**: configure the simulation to run from 0 to 5 s with a step size similar to 1 ms (or let the solver choose its own step). Enable logging or result export for the `omega` variable.
4. **Post-process and compare**: export the simulation results to CSV and compare the `omega`/`rpm` trajectory to the provided reference file ``. If your FMU and the FMU_Gateway are working correctly, the resulting curve should closely match the reference. Minor differences due to solver tolerance are acceptable.

Credibility and sources

This simplified model draws upon recognised turbocharger modelling practice. The fundamental relationship between turbocharger power balance and angular acceleration (Eq. (1)) is stated in a master's thesis on engine/turbocharger co-simulation. The same source discusses how turbocharger performance maps use reduced speed variables, which supports the selection of model parameters. Although Eq. (2) is a simplification, it preserves the qualitative behaviour of spool-up dynamics and is suitable for verifying that the FMU_Gateway correctly handles inputs, states and outputs.
