Tutorials

Tutorial 3 - Automotive Powertrain and Vehicle Simulation

Objective

This tutorial will lead you step by step to a powertrain model of varying complexity. The start will form a simple engine model. The model then is extended to containbasic powertrain components (clutch, gears, wheels), the car mass, and driving resistances in order to simulate processes such as the car starting up from zero speed. The third step of refinement leads to a model which allows to simulate different types of gearbox vibration phenomena. An outlook is given to further possibilities of developing more detailed models. It is assumed that you are familiar with the basic functionality of SimulationX.
Therefore, please refer to "Tutorial 1: Getting Started" for a general introduction on how to select elements from the libraries, how to connect them and enter parameters, how to run a simulation and how to open result windows.

- *Easy modeling of powertrain components*
- *Observation of results*
- *Extending of models*
- *Result analysis with parameter variation*
- *Influence of several parameters*

Part 1: Simulating Vehicle Acceleration

1. Engine Modeling

For accelerating a vehicle a very simple engine model will be sufficient. Usually an engine is described in terms of the torque as function of the engine speed. Create the SimulationX model shown in Figure 1.

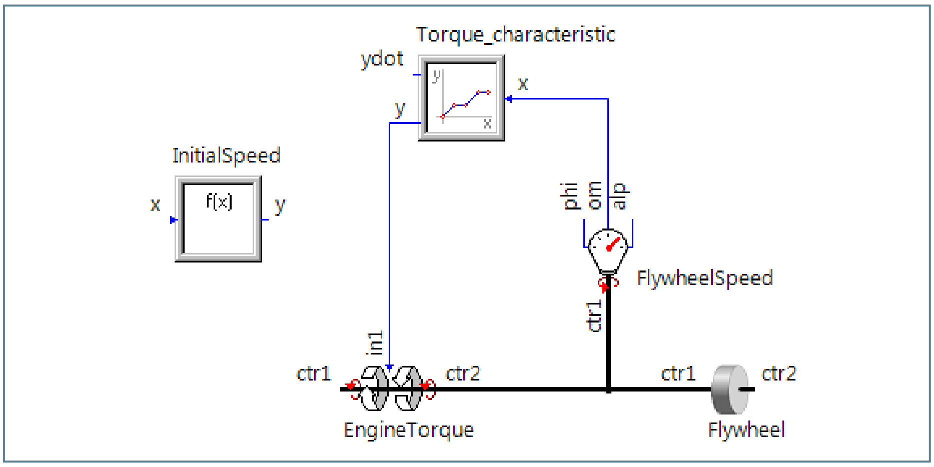


Figure 1: Simple Engine Model

To rotate a model element, select the element with L-Mouse and choose "Rotate Left" or "Rotate Right" from the "Elements" menu. You can also use the buttons  for this operation.

To change the label of an element double-click on it to open the properties window. Use the dialog page "General" to edit the name. Here you also have the possibility to position the label with respect to the object.

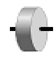
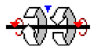
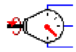
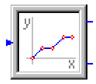
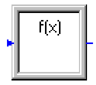
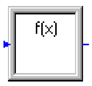
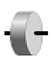
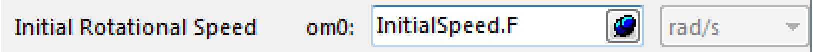
Number of elements	*Library name*	*Element name* *Function*	*Symbol*
1	*Mechanics/* *Rotational Mechanics*	*Inertia* *Inertia of flywheel, clutch and gearbox input shaft. Element for applying the engine torque and for measuring the engine speed*	
1	*Mechanics/* *Rotational Mechanics*	*External Torque* *Speed-depended engine torque*	
1	*Mechanics/* *Rotational Mechanics*	*Sensor* *Engine speed measurement(flywheel Sensor)*	
1	*Signal Blocks/* *Signal Sources*	*Curve* *Speed-torque characteristic*	
1	*Signal Blocks*	*f(x)* *Container for a model parameter, which is to be changed and is used in various model elements*	

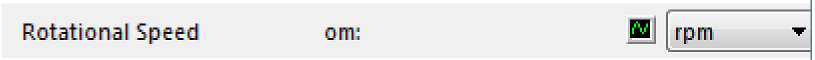
Table 1: Elements contained in the engine model

Once you have succeeded in creating the model structure according to Figure 1, you have to enter the parameters for the elements and have to activate the protocol attributes for the result variables which you want to plot after the simulation:

Model Object	*Parameter Input*
InitialSpeed 	*In "Initial Speed" we define a model parameter, which shall be easily accessible and which may be used in different model elements in the simulation. Because the function-element only has a base quantity, we have to define the used value by a string. This determines the unit of the parameter.* • *Set the function f(x) to 800 rpm* Function f(x) F: 800'rpm' - *Note that you can add a unit to a unitless parameter by attaching the unit name enclosed in single quotes.* • *Activate the protocol attribute for the Signal Output* Signal Output y: -
Flywheel 	• *Set the parameter "Moment of Inertia" to 0.35 kgm2 – a typical value for a passenger car engine* Moment of Inertia J: 0.35 kgm^2 • *Enter a reference to the parameter F of the element "InitialSpeed" (InitialSpeed.F) for the initial rotational speed of the flywheel*



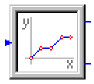
• *Activate the protocol attribute for "Rotational Speed" and change the unit of measurement to "rpm"*



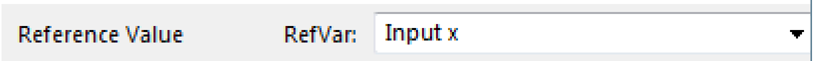
EngineTorque

• *Assign the torque of the source to its signal input by entering the name of the input (in1)*

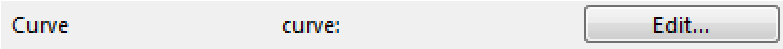
Torque_characteristic



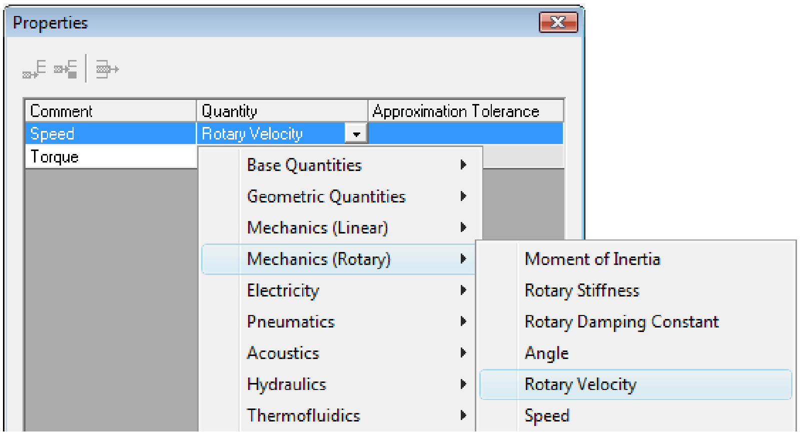
• *Set the "Reference Value" to "Input x" in order to make the result (torque) dependent on the input (engine speed)*



• *Open the curve dialog by click on the Edit button*



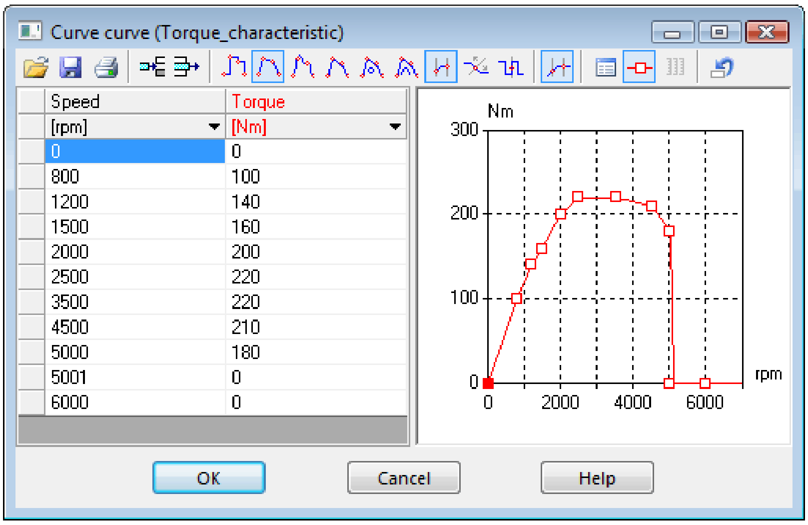
• *Define a name for the range and the domain of the curve, as well as the corresponding physical domains and units of measurement by clicking on the "Properties" – button* 



• *Select for x "Mechanics(Rotary)/Rotary Velocity" with a unit "rpm", and the name "Speed"*

• *Select for y "Mechanics(Rotary)/Torque" with a unit "Nm" and the name "Torque*

• *Insert the following values*



Speed [rpm]	Torque [Nm]
0	0
800	100
1200	140
1500	160
2000	200
2500	220
3500	220
4500	210
5000	180
5001	0
6000	0

• *Activate the protocol attribute for the "Signal Output"*

	Signal Output y: Nm

Table 2: Parameter Input

Now you are ready to perform a test run of your engine. In order to see it working open the result window for the flywheel speed (select “Result Curve.../Rotational Speed” from the context menu of the flywheel object) and start the simulation. You can see the speed increase until it reaches the maximum of 5000 rpm defined in the torque characteristic (see Figure 2).

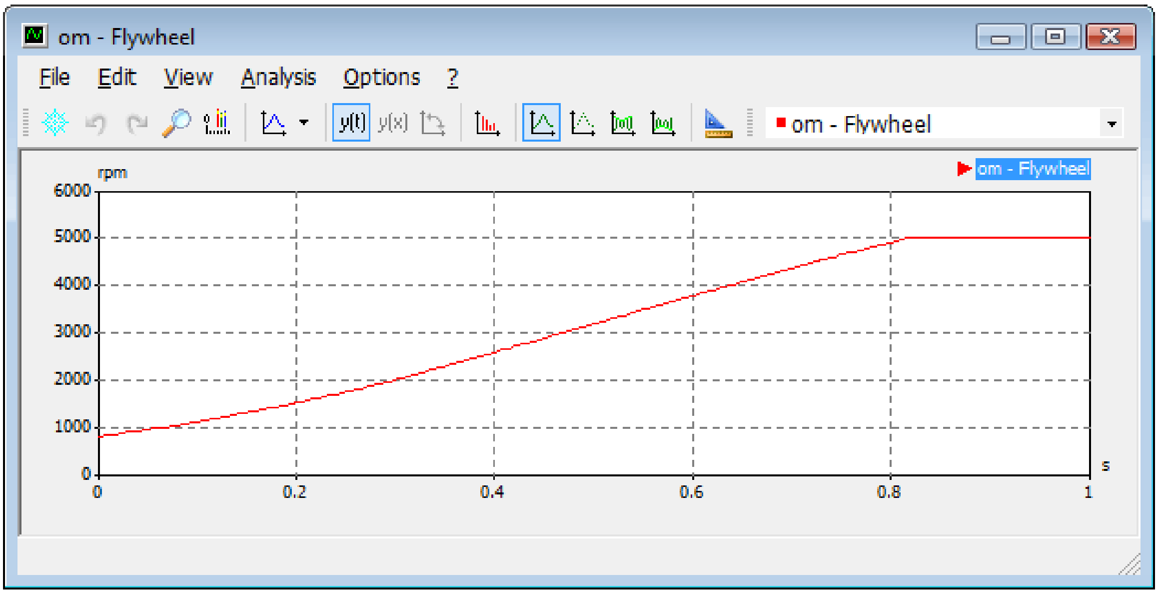


Figure 2: Engine run-up from 800 rpm to the maximum of 5000 rpm

2. Powertrain Modeling

You now proceed to the remainder of the vehicle – gearbox (with one fixed gear), axle gear, wheels, the mass of the car, and the driving resistance due to air drag and rolling friction. First reset your simulation, and then build your powertrain model as shown in Figure 3.

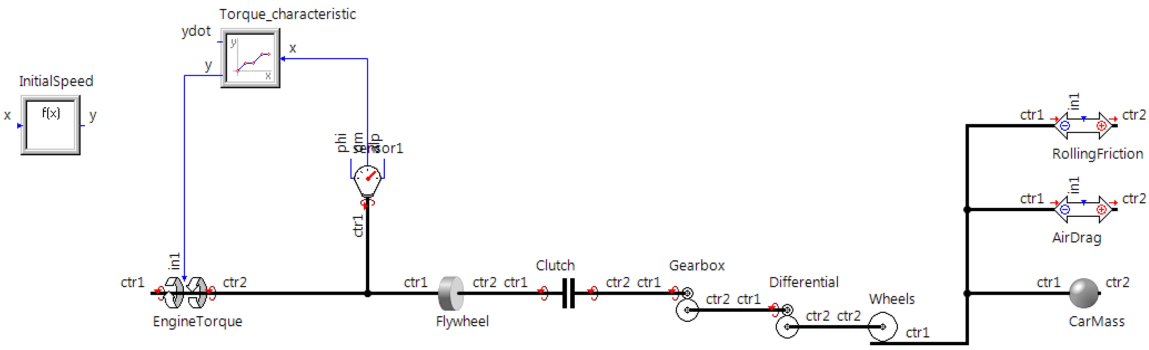
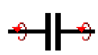


Figure 3: Complete powertrain model

Number of elements	*Library name*	*Object name Function*	*Symbol*
1	*Mechanics/ Rotational Mechanics*	*Rigid Friction* *Modeling the clutch* 	
2	*Mechanics/ Rotational Mechanics*	*Gear* *The transmission ratios for the selected gear and the differential gear*	

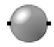
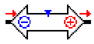
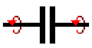
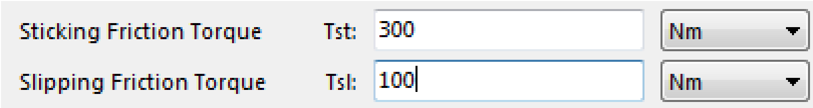
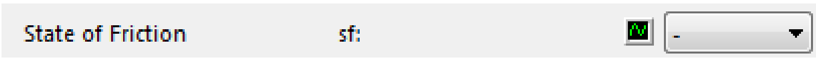
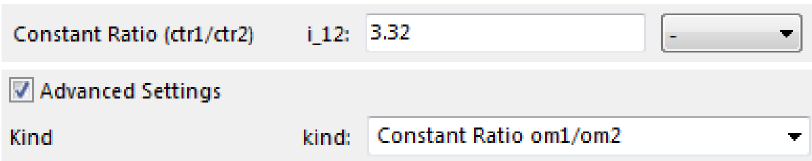
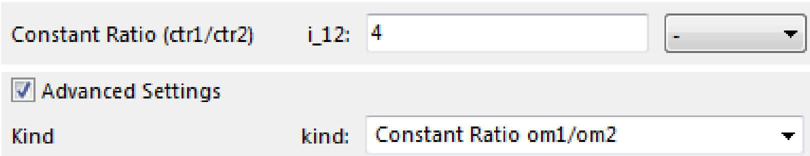
			
1	*Mechanics/ Rotational Mechanics*	*Rotational-Linear Transformation* *The Wheels* 	
1	*Mechanics/ Linear Mechanics*	*Mass* *The car mass* 	
2	*Mechanics/ Linear Mechanics*	*External Force* *Driving resistance – air drag and wheel friction* 	

Table 3: The following new objects are contained in the model

You can rename the objects as usual. Next the new model objects are parameterized.

Model Object	***Parameter Input***
Clutch 	• *In this rotary friction element the torque in the sliding state and the torque necessary to break loose from the sticking state have to be specified. The break-loose torque should be greater than the maximum torque deliverable by the engine, so we set it to 300 Nm. The sliding torque is set to a smaller value. Choose 100 Nm in order to make it as large as the torque of the engine at initial speed (800 rpm). Thus the engine speed will remain constant until the clutch is fully closed.*  • *Activate the protocol attribute for the result variable "State of Friction" in order to observe the behavior of the clutch element during simulation.* 
Gearbox 	• *In "Gearbox" the gear ratio for the selected gear has to be given. We select the transformer such, that the gear ratio calculates as the ratio of the rotary speeds (kind is "Gear Ratio om1/om2"):* $$i = \frac{\omega_2}{\omega_1}$$ *The indices correspond to the respective connectors. The connector with the red arrow is connector 1. Assume that the car has a gear ratio of 3.32 (83 and 25 teeth) in first gear, then the following is entered in the parameter dialog:* 
Differential 	• *The differential gear ratio is selected in the same way as the gearbox gear ratio. Take the ratio 4, which is reasonable for a differential gear.* 
Wheel 	• *Here we have to specify the translation of the rotary motion of the drivetrain to the translatory motion of the car. This transformation is performed in the wheels. As for the gears, the ratio of the speed at the translatory connection (connected to the car mass) to the speed of the rotary connection (connected to the differential) has to be given. This is the ratio of wheel circumference (in meters) and the corresponding angle 2π (in radians), i.e.*

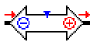
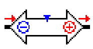
<table>
<tr>
<td></td>
<td>
the wheel radius. Selecting a radius of 0.35 m the following has to be entered:
Constant Ratio (ctr1/ctr2) i_TR: 0.35 m/rad
☑ Advanced Settings
Kind kind: Constant Translational-Rotational Transmission (x/phi)
Note, that in our arrangement (Figure 3) the wheel is built in "in reverse", connector 2 points towards the engine side, connector 1 towards the mass.
</td>
</tr>
<tr>
<td>CarMass
</td>
<td>
• The car shall have a mass of 1400 kg, which is entered in the parameter dialog of the CarMass element
Mass m: 1400 kg
The initial values for displacement and speed should be set to zero (and fixed using the blue pin in order to prevent the solver from altering it during the initial value calculation) in order to simulate a start form zero speed and displacement.
• Activate the protocol attributes for the result values "Velocity" and "Acceleration" and set the unit of measurement to km/h for the velocity.
Velocity v: km/h
Acceleration a: m/s²
</td>
</tr>
<tr>
<td>AirDrag
</td>
<td>
• When moving a car observes two important resistance forces: The air drag and the rolling friction. The air drag calculates as
$$F_D = C_d * A * \frac{\rho * v^2}{2}$$
$C_d = 0.31$ - air drag coefficient, $A = 2.2m^2$ - projective area of the car,
$\rho = 1.199 \frac{kg}{m^3}$ - air density, v - vehicle speed
Above formula can be entered directly into the parameter for the force. A special property is exploited in order to derive the velocity needed for air drag calculation. SimulationX provides the motion state variables (displacement, speed, acceleration) as any other system parameter or variable. Using the corresponding element name they can be accessed. In our example model we use the speed CarMass.v When entering the values, note that all of them have to be given in SI base units (which is the case here)
Force F: 0.31*2.2*1.199*CarMass.v N
0.31*2.2*1.199*CarMass.v^2/2
</td>
</tr>
<tr>
<td>RollingFriction
</td>
<td>
• The second driving resistance is the force resulting from rolling friction, which is almost constant over wide ranges of vehicle speeds. It is calculated as
$$F_R = \gamma_R * m * g$$
$\gamma_R = 0.01$ - rolling friction coefficient (asphalt road),
$m = 1400 kg$ - car mass,
$g = 9.80665 \frac{m}{s^2}$ - gravity.
Again, we have to enter all values in their SI base units
Force F: 0.01*gravity*CarMass.m N
</td>
</tr>
</table>

Table 4: Parameter input

At this state, save the model, so that you can reuse it in Part 2 of the tutorial.
Now acceleration tests can be performed. Open the result windows for the clutch friction state, the car mass speed, and the car mass acceleration. Set the simulation "Stop Time" (Menu "Simulation/Settings") to 5 s and "Start" the simulation.

You will see the curves shown in Figure 4 and Figure 5.

The curves are currently displayed in four windows. In order to form joint displays, click the colored marker of the curve in the legend (upper right corner of the plot), drag it to the destination window (another result display), and drop it by releasing the mouse button.

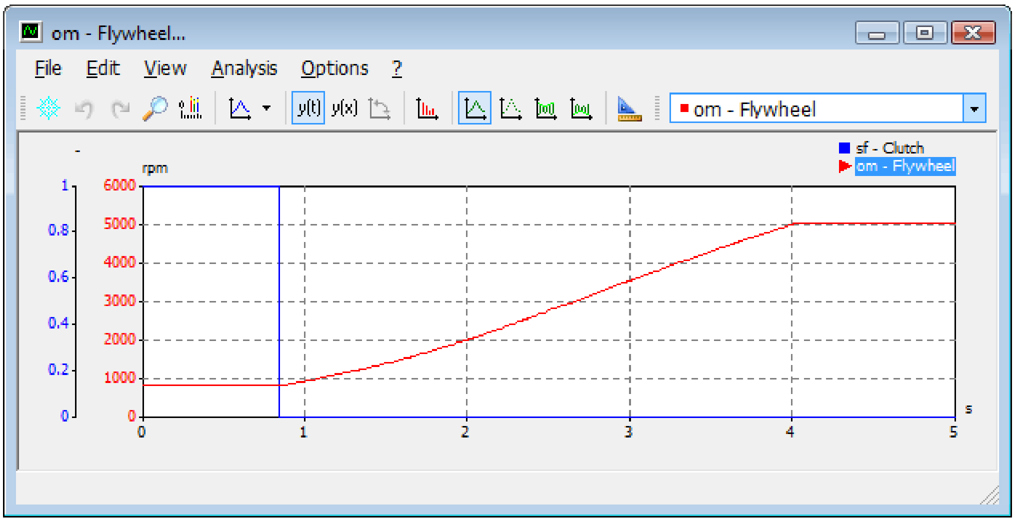


Figure 4: Clutch state and engine speed

The engine speed remains constant until the clutch is fully closed. Since you made the slipping torque as large as the initial engine torque, the engine torque is fully used for car acceleration via the slipping clutch and thus the engine remains at constant speed. After the clutch is engaged, the car accelerates until the maximum engine speed is reached.

Figure 5 shows the car acceleration from zero speed to about 50 km/h. In the initial phase the clutch is slipping and the car accelerates with constant acceleration. With the clutch closed the acceleration first drops, since not only the vehicle mass, but also the flywheel inertia has to be accelerated now (the engine gains speed). Acceleration stops when the engine speed has reached its maximum.

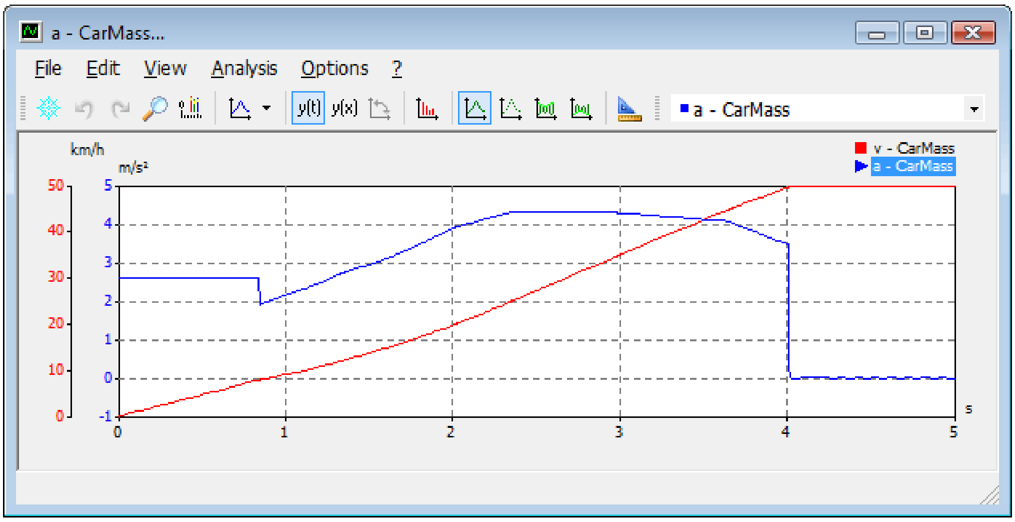


Figure 5: Car velocity and acceleration

Next you will observe the acceleration in higher gears. As in real life higher gears are only selected with the vehicle in motion at a high speed, so an initial speed has to be given to the car mass. In order to simulate this acceleration, change the following parameters:

Model Object	***Parameter Input***
Gearbox 	• *You now assume being in e.g. 4th gear with a gear ratio 0.97:* Constant Ratio (ctr1/ctr2) i_12: 0.97 -
CarMass 	• *You will let the car start at 100 km/h – do not forget to change the unit of measurement accordingly.* Initial Velocity v0: 100 km/h

Table 5: Parameter changes

Now reset the simulation, set the simulation “Stop Time” (Menu “Simulation/Settings”) to 30 s and “Start” the simulation again.

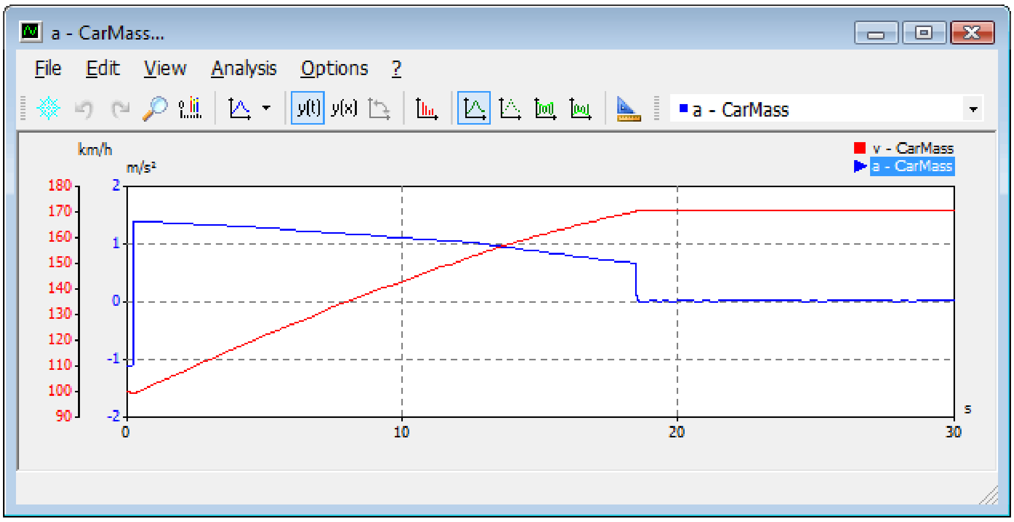


Figure 6: Car velocity and acceleration when changing gears at 100 km/h

As one would expect, acceleration is lower than in the first example with the low gear. Once the top speed for the gear is reached (here 170 km/h), acceleration ceases.
The last experiment performed with this example is the study of the air drag influence.

Before starting, freeze the speed and acceleration curves by pressing the  button in the toolbar of the result windows.

Model Object	***Parameter Input***
Driving resistance *(Air drag)*	• *Change the air drag coefficient to a significantly higher value, we take 0.36 as an example* Force F: 0.36*2.2*1.199*CarMass.v N

Table 6: Now the air drag is increased

After restarting the simulation you can observe the impact of the increased air drag – a slightly lower acceleration, which causes the top speed to be reached almost 2s later.

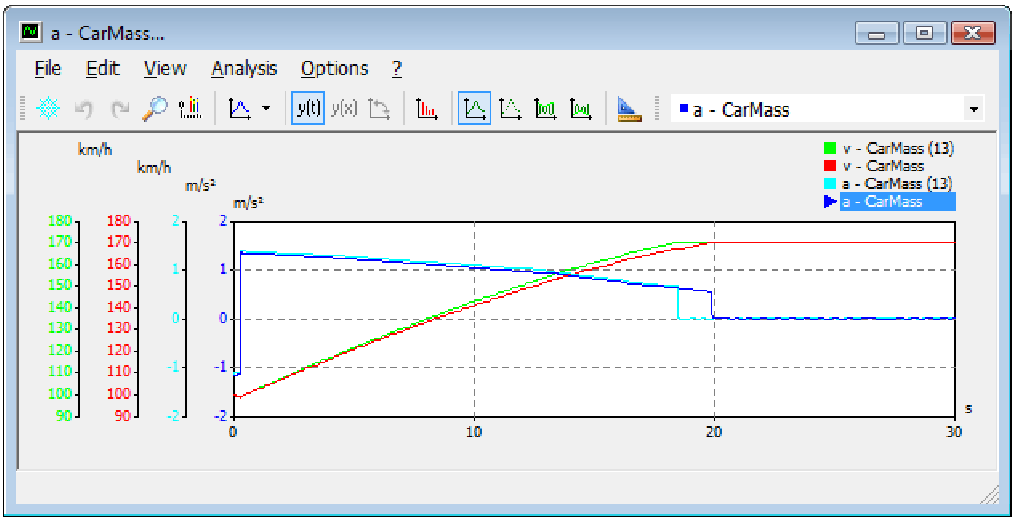


Figure 7: Comparison of different air drag coefficients

Part 2: Simulating Gearbox Noises

In the second part of the tutorial we will turn our attention to a more specialized problem – the analysis of gearbox noises. There are two types of noise in the gearbox, which shall be studied here

- *Rattling of teeth*
- *Whining of gearbox teeth under load.*

The study of such effects requires a greater detail in the model. First you have to use another engine model. Gearbox noises are excited by the unevenness of the engine rotation, which is caused by the changes in the cylinder torque during compression and combustion. For the gearbox noises you need a gear model which incorporates backlash (responsible for rattling) and considers gear stiffness and gear meshing (the gear meshing can excite higher frequencies in the system which can be audible as gearbox whining).

1. Teeth Rattle Simulation

You start from the model you saved earlier in Part 1 and replace some of the model components. Before you start you can close the result windows in order to tidy up your workspace. Also, reset the simulation first so that you can edit your model. Rearrange the original model (shown in Figure 3) such, that you obtain the new model shown in Figure 8. You assume the gearbox to be a one-stage gearbox (i.e. there is only one gear ratio between gearbox input and output shafts) as it is e.g. used in vehicles with front-wheel drive.

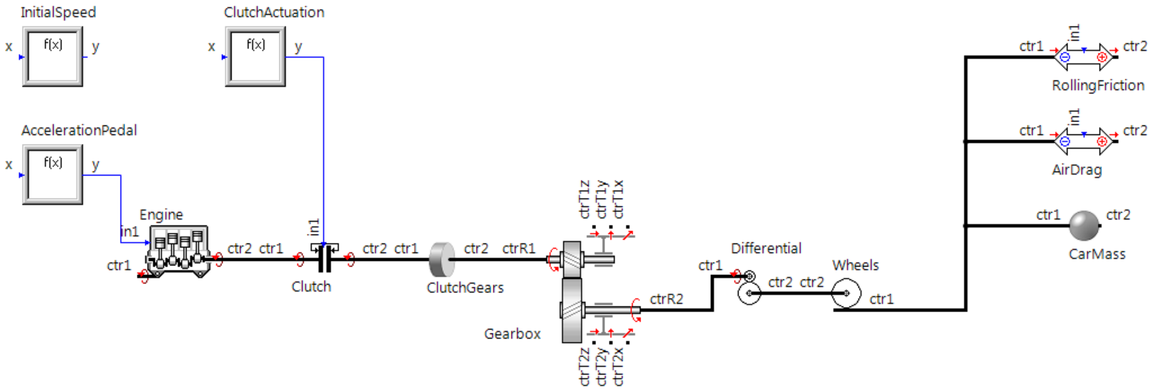


Figure 8: Refined powertrain model

The following new elements appear in the model:

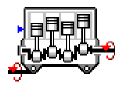
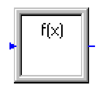
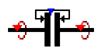
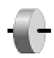
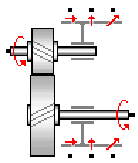
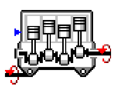
Number of Elements	***Library name***	***Object name Function***	***Symbol***
1	*Power Transmission/ Motors and Engines*	*Combustion Engines* *Engine model describing the influence of the combustion process in each cylinder of the provided torque by typical normalized functions*	
2	*Signal Blocks*	*f(x)* *Provision of accelerator pedal and clutch pedal signals to be fed into engine and clutch*	
1	*Power Transmission/ Couplings and Clutches*	*Disc Clutch* *Pedal-operated single-disk dry clutch*	
1	*Mechanics/ Rotational Mechanics*	*Inertia* *Inertia of clutch disk and gearbox input; auxiliary element for connecting the two elements Clutch and Gearbox*	
1	*Power Transmission/ Transmission Elements*	*Gear* *Detailed model of a gear contact including stiffness, damping and backlash*	

Table 7: New Elements

You can rename the new objects, as you like, in the sequel they will be called by the names shown in Figure 8.

As usual, the model has to be parameterized – the new components have to be adapted and some of the existing have to be modified. The required modifications are listed in the following table:

Model Element	***Parameter Input***
Engine 	• *For the parameterization of the engine you need to know the nominal power, nominal speed and the number of cylinders. The nominal values appear at the points of maximum power output of the engine. From the torque characteristic used previously you can calculate the power curve by simply multiplying engine speed and torque.* *!! Do not forget to transfer the engine speed to rad/s units before the calculation !!*

One observes that the maximum power output is reached at 4500 rpm and is about 99 kW. The engine shall have 4 cylinders. So you enter in the parameters dialog page 1:

Injection	inj:	in1	-
Nominal Power	Pn:	99	kW
Nominal Speed	omn:	4500	rpm
No. of Cylinders	nz:	4	-

Note that the limitation to 5000 rpm used in the simple model was an arbitrary assumption about the engine control and the engine element used here runs up to higher speed. In order to achieve the same behavior a maximum speed control could be implemented. For the experiments performed in the sequel this is not essential so you omit this.

In parameters dialog page 2:

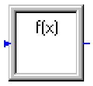
- *Set the engines torque to 0.35 kgm2 (before in flywheel)*

Inertia	J:	0.35	kgm^2

- *Link the initial speed to "InitialSpeed.F"*

Initial Speed	om0:	InitialSpeed.F	rad/s

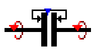
AcceleratorPedal



The engine element behavior is controlled by a normalized signal between 0 (no acceleration signal) and 1 (full acceleration). You will perform our experiment with full acceleration, i.e., a 1 is entered in the parameter dialog:

Function f(x)	F:	1	-

Clutch



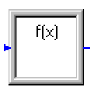
For the clutch parameterization you select typical values of a passenger vehicle clutch, which matches to the torque output of the engine. The clutch shall be a single-disk dry clutch, i.e., you have 2 friction surfaces. Reasonable settings for the remaining parameters are:

- *Outer diameter: 220 mm*
- *Inner diameter: 130 mm*
  - *Static friction value: 0.8 (this is higher than typical values, but we do not want the clutch to slide due to torque peaks once it is closed)*
- *Sliding friction value: 0.2*
  - *Press-on force: 8000 N*

All other values remain at their default settings as indicated in the parameter window. The parameters are set in the "Parameters" section of the window:

Switching Signal	sw:	in1	-
Friction Surface Outer Dia.	da:	220	mm
Friction Surface Inner Dia.	di:	130	mm
Disk Thickness	tD:	2	mm
No. of Friction Surfaces	ns:	2	-
Static Friction Coefficient	mu0:	0.8	-
Sliding Friction Coefficient	mu:	0.2	-
Press-On Force	Fp:	8000	N
Force Buildup Time	tu:	0.2	s
Friction Materials	kindM:	Steel - Sintered Bronze	

ClutchActuation



As well as the engine, the clutch is actuated by a normalized signal – 0 for open and 1 for closed. If the signal changes from 0 to 1 the clutch closes automatically, controlled by the setting of the "Force Engaging Time" parameter of the clutch (we left it at the default setting). In order to allow the engine to gain some speed before the clutch is closed, you switch the actuation signal at 0.1s. The easiest way to perform this is the exploitation of a logical expression. The expression if t>0.1 then 1 else 0 elegantly defines a signal, which starts at 0

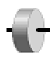
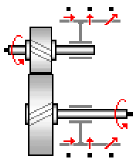
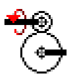
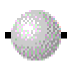
	and changes its value to 1 at time 0.1s. This is entered in the parameter dialog of the signal block Function f(x) F: if t>0.1's' then 1 else 0 -
Clutch_Gears 	•*Set the parameter "Moment of Inertia" to 0.01 kgm2 (representing the clutch disk and the engine-side gearbox parts, but being mainly an auxiliary node):* Moment of Inertia J: 0.01 kgm² •*Verify that "Initial Velocity" is set to zero:* Initial Rotational Speed om0: 0 rad/s
Gearbox 	•*In this model object you can set a multitude of parameters and have a variety of choices how the stiffness, damping, and tooth engagement are specified. For our simple experiment you leave all parameters at their defaults apart from the number of teeth for the two gear wheels and the backlash. These numbers are selected that they result in exactly the same gear ratio as used for the first gear in the model in Part 1 of the tutorial:* *– No. of Teeth, Wheel 1: 25* *– No. of Teeth, Wheel 2: 83* *Further, specify a backlash of 0.1 mm* Normal Modulus mn: 3 mm Helix Angle beta: 0 ° Common Face Width bw: 15 mm Rotary Backlash jt: 0.1 mm No. of Teeth (Wheel 1) z1: 25 - No. of Teeth (Wheel 2) z2: 83 - Pressure Angle alphan: 20 ° •*Make sure, that the "Consideration of Stiffness Change" is not marked. This option will be used in a later experiment.* ☐ Consideration of Stiffness Change •*Now go to the "Results" page of the parameter window and enable the protocol attribute for the normal forces on the tooth surfaces* Normal Force, Right Fl... Fbnr: N Normal Force, Left Flank Fbnl: N
Differential 	•*The detailed gearbox object behaves like a real set of gears. Consequently, the output shaft rotates in reverse direction compared to the input shaft. This was not the case in the simple vehicle model, i.e. with the new setup the vehicle would actually move backward. Since the air drag and rolling resistance settings only hold for a forward movement, you have to compensate for the inverted direction of rotation.* *This is conveniently achieved in the differential by changing the sign of the gear ratio* Ratio om2/om1 i_21: -4 -
CarMass 	•*Make sure that you reset the car's initial velocity to zero (it might have another value from earlier experiments)* Initial Velocity v0: 0 m/s *and that the protocol attributes for speed and acceleration are enabled* Velocity v: km/h Acceleration a: m/s²

Table 8: Model Element Parameter Input

You are now ready to run your simulation. For this set the simulation stopping time to 5 s and the "Min. Output Step..." (dtProtMin) to 0.0001 s. This will ensure that higher frequency components get displayed properly too.

Open the result windows for the "Rotatory Speed" of the engine and the teeth forces of the gearbox. After running the simulation you will see the results displayed in Figures 9 to 10.

It is clearly visible, that the engine speed first increases until the clutch starts to close. Then the speed is decreased again until the clutch is fully closed, whereby the car starts to move due to the friction torque transmitted by the clutch. Once the clutch is fully closed, the whole setup accelerates. Looking at the forces at the teeth one observes that there is a normal force on the left as well as on the right tooth surface. Since there is a backlash in the gearbox, this means, that the gears rattle. Rattling starts first after the clutch is closed and then shows a resonance at about 2700 rpm. It ceases above engine speeds of 3300 rpm.

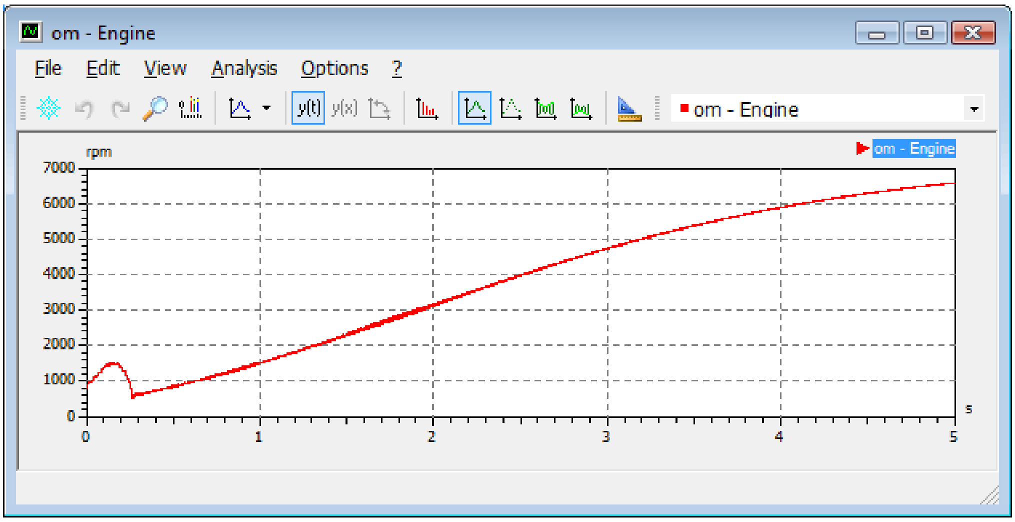


Figure 9: Flywheel speed

Let us now have a closer look at the rattling process. For this track the marker in the legend from one of the force windows into the other in order to form a joint display of the two forces. Next enlarge the view that you can see a section in the resonance range. Either click the magnifying glass and zoom into the desired area or choose the "Settings" dialog, go to the "X-axis" window, deselect the option "Automatic Scaling", enter 1.8 s and 1.83 s as "Min" and "Max" values, and set the number of "Ticks" to 3.
Now you see the curves shown in Figure 12. Obviously, the gears are in contact (i.e. there is a normal force) intermittently. In-between there is a period where the gears rotate relative to each other and there are no normal forces at all since there is no contact.

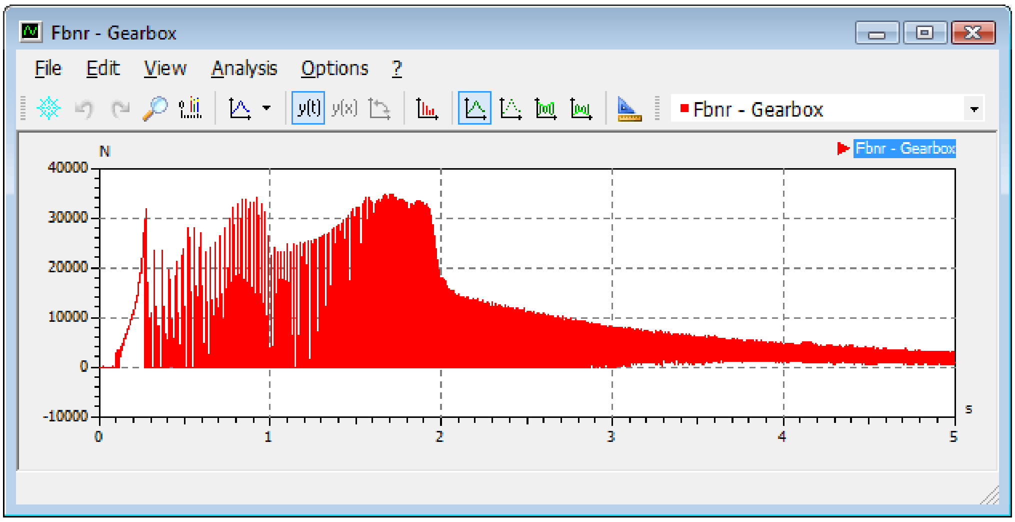


Figure 10: Normal force to the right tooth surface (driving side)

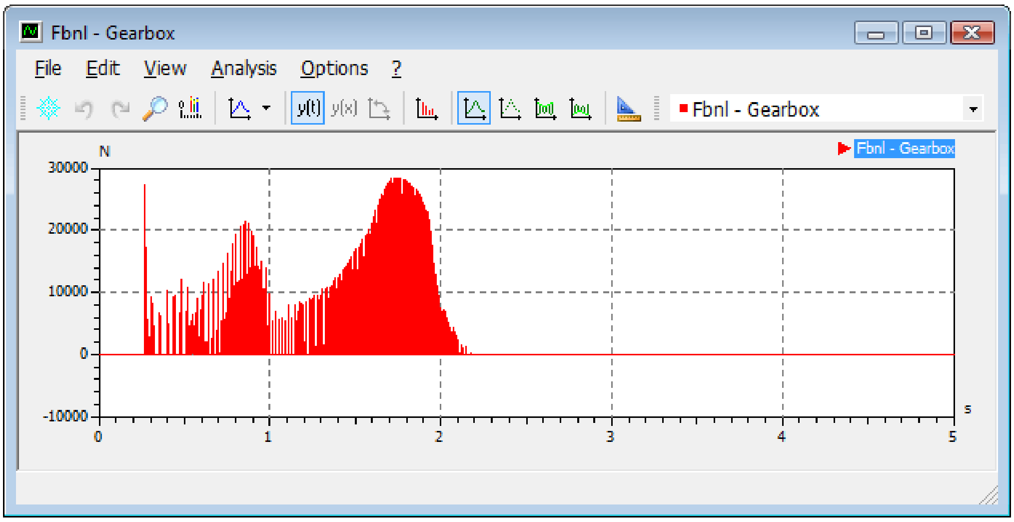


Figure 11: Normal force to the left tooth surface

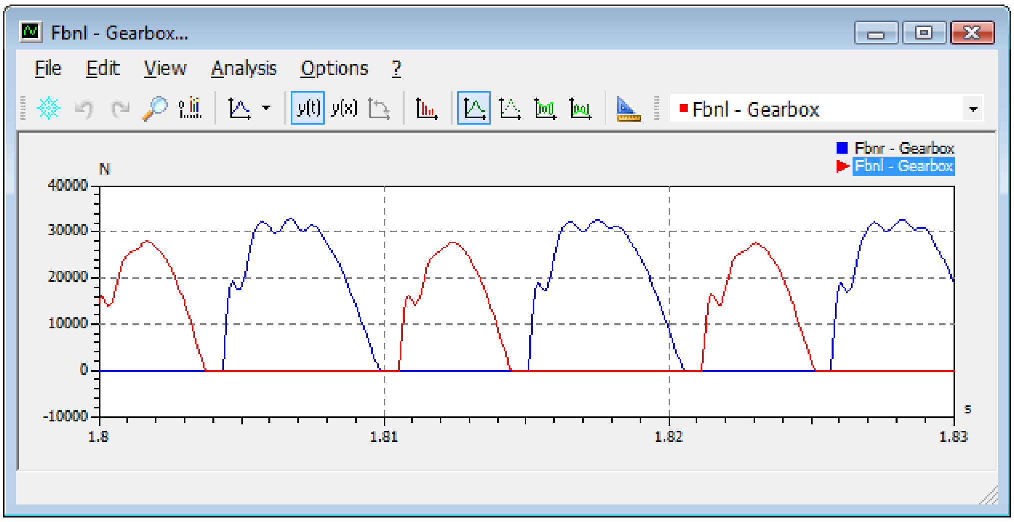


Figure 12: Normal forces in higher resolution, showing teeth rattle

Looking at the period of the force signals (the repeating peaks are 11 ms away from each other) you can easily identify the reasons for the teeth rattle. Since you have a four-cylinder engine, there are two ignitions per revolution. At about 2800 rpm these impulses appear at a frequency of 93 Hz and thus are about 11 ms away from each other.

You can measure curve points (and thus periods in this example) by selecting the desired curve in the legend (it gets marked with a triangle) and clicking with the left button into the diagram and holding it. A hair cross appears, which has the same x-coordinate as the mouse pointer and is placed on the curve. The current point coordinates are shown in the toolbar. Moving the mouse you obtain the coordinates of other curve points.

This is an example of an unwanted gearbox behavior. In a car design using SimulationX you would have spotted the problem at an early design stage and would have been able to take appropriate counter-measures.

2. Analysis of gearbox whining

You will now make a change to the gearbox, which allows observing another type of gearbox noises – the whining – which is excited by the teeth meshing. In order to perform this simulation, you leave the model as it is and change only one parameter:

Element	*Parameter Input*

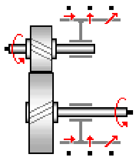
Gearbox 	• *Mark the checkbox for the "Consideration of Stiffness Change" in the parameter dialog* ☑ Consideration of Stiffness Change *This will lead to a stiffness, which depends on the actual number of teeth engaged and the degree of engagement. This stiffness variation is periodic with the gear meshing frequency and can lead to the excitation of vibrations in the powertrain.*

Table 9: Changes in gearbox parameters page "Paramters"

For the simulation you have to scale the output window (as used in Figure 12) along the force direction. Disable the "Automatic scaling" on the "Y-axis" page of the "Settings" dialog and enter a maximum force of 60000 N. The number of "Ticks" has to be changed to 7 accordingly.

Reset your simulation now and run it again. After a while you will see the force curves of the selected time interval appearing in the result window. The result is displayed in Figure 13.

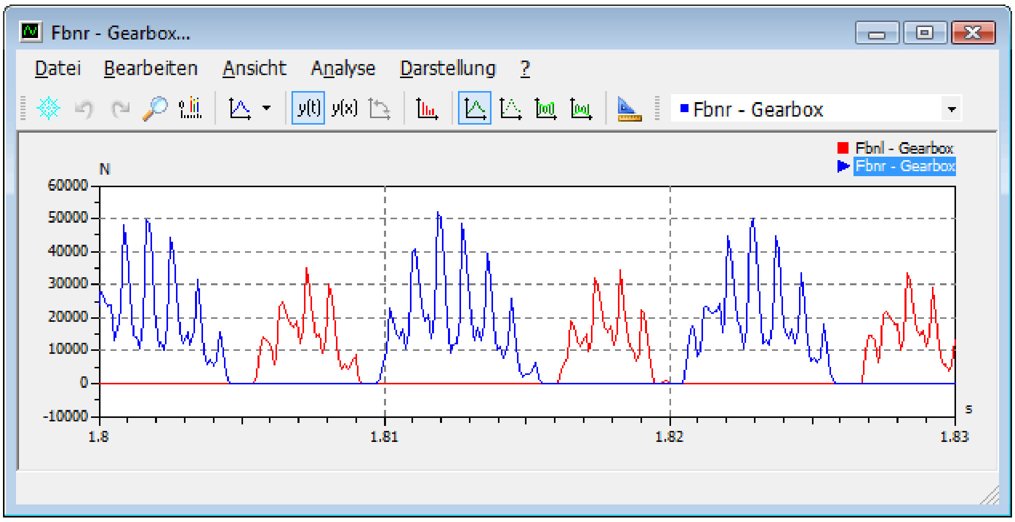


Figure 13: Normal forces in higher resolution showing teeth rattle and whining

Obviously, there is now a component with a considerably higher frequency. Measuring the distance between neighboring peaks one finds approximately 0.85 ms, i.e. 1.18 kHz as the fundamental frequency. In order to confirm that the gear meshing causes it, you can calculate the meshing frequency, which is 25 teeth per engine rotation. It gives at an engine speed of 2800 rpm about 1.18 kHz.

Conclusion

Now you are free to experiment with the simulation models and change parameters or extend the models to greater detail. This might include inertia, stiffness, and damping of different shafts in the powertrain, the description of tire behavior, or a detailed engine model reflecting the inertia, stiffness, and damping of the crankshaft, the mass forces of the piston, and models for the combustion process. You can build many detailed elements such as a Cardan shaft or various detailed cylinder and engine models with the TypeDesigner and can be added to the type collection in the SimulationX libraries. So you have the flexibility to make your model as detailed as required for your simulation and analysis tasks.

Let us finally resume the goals, which you should have reached in this tutorial

- *You now know how to construct car and powertrain models for different simulation goals. The more complex model elements usually come with a set of default parameters, which you often can leave unchanged. Nevertheless, you should always think about the correctness of the parameters you use in order to ensure, that the model accurately maps the reality.*
- *You have learned about various methods to parameterize objects – not only numbers, but also mathematical formulas and logical expressions. Further, you now know how to access other variables and parameters by referencing to the corresponding quantities.*

- *You are able to use signal blocks for making multiply used variables available, such as common parameters of several objects, and you know how to assign physical meanings and units to signals.*
- *You have seen that the amount of detail in the model grew with the complexity of the effects, which were to be observed. This is worth a consideration in every simulation. Growing complexity in a model increases the overall simulation time. So the model should always be only as complex as it is necessary in order to map the effect to be observed.*
- *SimulationX is a tool for intuitive system simulation, since models with mixed physical domains (in our examples mechanics and control systems) can be built very quickly.*
- *The object-oriented approach of SimulationX allows you to modify a given model structure very easily and adapt it to new tasks and specifications.*
- *SimulationX models clearly reflect the structure of the underlying physical systems, so you always see your real world problem when looking at the model.*