



PLECS

Tutorial

Modeling an Electric Vehicle Using the PLECS Mechanical Domain

Last updated in tutorials release 1.0

www.plexim.com

- ▶ Request a PLECS trial license
- ▶ Check the PLECS documentation

1 Introduction

In this exercise you will learn how to use the mechanical modeling features of PLECS to create a combined electro-mechanical system representing an electric vehicle propelled by an electric motor. The specific learning goals for this exercise are listed as follows:

- Familiarize yourself with the Rotational and Translational libraries of the PLECS Mechanical Domain. Each of these libraries consists of Sources and Components subcategories used to model rotational and translational mechanical dynamics. Additionally, each library has a Sensors category to measure mechanical variables of interest such as speed, position, and force/torque.
- Learn which components are needed for modeling the motor and the electric driveline of the vehicle. Develop a mechanical model for the motor and rotating driveline.
- Develop a translational vehicle model to observe the effects of a lumped vehicular system experiencing a road load force and tire frictional force.

2 Motor and Drivetrain Shaft Modeling

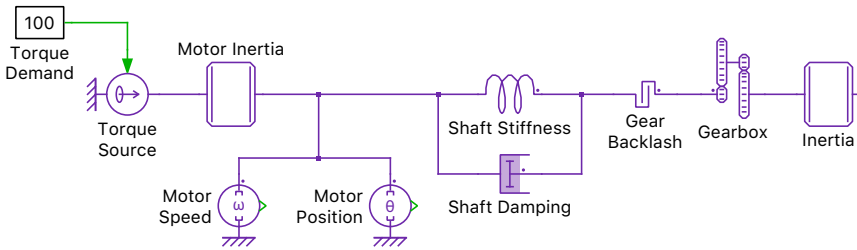


Figure 1: Motor and drivetrain

The motor, used to propel the electric vehicle, can be modeled in a simple way as an ideal torque source and a rotating mass using the Inertia block. The dynamics of the motor are given by:

$$\frac{d\omega}{dt} = \frac{1}{J_{\text{motor}}} \sum \tau, \quad (1)$$

where the torque, τ , is positive into the body.

A driveline shaft connects the motor to a gearbox. The dynamics of the shaft are modeled using two PLECS components in parallel. A Torsion Spring block models the shaft stiffness and a Rotational Damping block models the shaft damping losses. The gearbox is used to connect a fast rotating shaft to a slower rotating shaft to reduce the motor speed to the wheel speed. The characteristics of the gearbox are given by:

$$\omega_{\text{fast}} = g \omega_{\text{slow}} \quad \text{and} \quad \tau_{\text{fast}} = \frac{\tau_{\text{slow}}}{g}, \quad (2)$$

where g is the gear ratio.

Care must be taken when connecting the gearbox component as the flange connected to the faster-moving shaft is marked with a dot (assuming a gear ratio of greater than 1). Additionally, a Rotational Backlash component can be used to model any gear non-ideal backlash. A model for the motor and driveline shaft is shown in Fig. 1.

The angular speed and rotor position of the driveline shaft can be measured by using a Rotational Speed Sensor block and an Angle Sensor block, respectively. Each rotational (translational) mechanical sensor block can make measurements either with respect to the rotational (translational) reference frame or between the two flanges of the sensor. When measuring quantities between the two flanges, a measurement is made at the flange marked with a dot with respect to the unmarked flange.

**Your Task:**

- 1 Model the motor using an Inertia block from the component library “Mechanical” in the sub-libraries “Rotational” and “Components”.
- 2 Add a Torque (Controlled) block and connect a Constant block to its input to produce 100 Nm of torque that is demanded by the driver. Double-click on the torque source and select Connected to reference frame from the drop-down menu to produce a torque relative to the rotor reference frame. Connect the controlled torque source to the motor.
- 3 Model the shaft stiffness and damping using the Torsion Spring and Rotational Damper blocks, respectively. Connect these components to the motor, as shown in Fig. 1.
- 4 Add the gearbox using the Gear component. Connect in the gear backlash between the shaft and the gearbox using the Rotational Backlash block.
- 5 Enter the parameters for the motor model using the values given in Table 1.
- 6 Measure the driveline quantities with respect to the rotor reference frame by double-clicking on each sensor and selecting Connected to reference frame from the drop-down menu. Connect these to the driveline shaft.
- 7 The natural frequency of the driveline, f_n , is also of interest. Assuming the moment of inertia of the vehicle to be large with respect to the motor we can calculate the natural frequency of the driveline by solving the differential equation that arises due to the shaft stiffness and damping. The formula for the natural frequency of the driveline is given by:

$$f_n = \frac{1}{2\pi} \sqrt{\frac{c}{J_{\text{motor}}}}, \quad (3)$$

where c is the spring constant and J_{motor} is the moment of inertia of the motor. (~ 9 Hz)

- 8 Connect an Inertia block to the secondary side of the gearbox with a value of $10^4 \text{ kgm}^2/\text{rad}^2$ to represent the inertia of the vehicle. This will dampen the system such that the driveline oscillation subsides after only a few cycles. Connect a Scope to the Rotational Speed Sensor block and run the simulation using the default settings to monitor the motor speed. Using the data cursors, measure the period for which one full oscillation occurs. (~ 0.116 s) Calculate the natural frequency from this period and compare this value with the calculated frequency from the previous step.



At this stage, your model should be the same as the reference model, `mechanical_domain_1.plecs`.

3 Vehicle Modeling

3.1 Model the vehicle and forces

To model an electric vehicle using PLECS, two components are needed: mass and friction. The Translational Mass block absorbs the force from the wheel axle connected to its terminals and the speed of the mass is determined by integrating the equation:

$$\frac{dv}{dt} = \frac{1}{m} \sum F \quad (4)$$

where the force, F , is positive into the body.

Table 1: Motor and driveline shaft parameters

Component	Parameter	Value
Motor Inertia	Moment of inertia	0.014 kgm ² /rad ²
	Initial speed	0 rad/s
	Initial angle	0 rad
Shaft Stiffness	Spring constant	45.45 Nm/rad
	Equilibrium (unstretched) displacement	0 rad
	Initial deformation	0 rad
Shaft Damping	Damper constant	0.5 Nm/rad
	Initial displacement	0 rad
Gearbox	Gear ratio	10
Rotational Backlash	Total backlash	0 rad
	Initial displacement	0 rad

The Translational Friction block is used to model both the tire friction and the road load experienced by the vehicle. In the Translational Friction block the “Breakaway friction force” parameter is the maximum magnitude of the stiction force. Additionally, the Translational Friction block models the Coulomb friction force and the viscous friction force with respect to a translational reference frame. These forces are exerted on the wheel axle and therefore are connected to the wheel axle in parallel to the vehicle mass, as shown in Fig. 2.

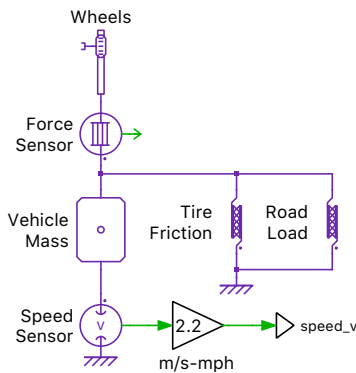


Figure 2: Simple electric vehicle model

 **Your Task:**

- 1** Remove the Inertia block connected to the secondary side of the gearbox as well as the Scope.
- 2** Model the vehicle using a Mass block from the component library “Mechanical” sub-libraries “Translational” “Components”. From the Sensors list, add a Translational Speed Sensor block. Measure the speed of the vehicle relative to a translational reference frame by double-clicking on the sensor and selecting Connected to reference frame from the drop-down menu.
- 3** Measure the vehicle translational speed using the Translational Speed Sensor block, again with respect to the translational reference frame. Convert the measured speed from meters per second (m/s) to miles per hour (mph) using a Gain block and provide a gain value of 2.2. Monitor the vehi-

cle speed in a PLECS Scope along with angular speed and rotor position of the driveline shaft.

- 4 Model the tire friction and road load using two Translational Friction blocks from the Components list. Use the Translational Reference block to produce the tire friction and road load forces relative to a translational reference frame.
- 5 Enter the parameters for the vehicle model using the values given in Table 2. Note that the road load values used represent a vehicle driving up a steep hill.
- 6 To integrate the rotational components and translational components of a vehicular system, they should be coupled to each other through the wheels of the vehicle. The vehicle wheel is modeled using the Rack and Pinion block from the component library “Mechanical” in the sub-libraries “Translational” and “Components”. Connect the rotational driveline to the translational vehicle model through a Rack and Pinion block. Set the “Pinion radius” to be 0.28 m.
- 7 From the Sensors list, add a Translational Force Sensor block to measure the force transmitted from the wheel to the translational vehicle model. Monitor the force in the same PLECS Scope as the other signals.

Table 2: Electric vehicle parameters

Component	Parameter	Value
Vehicle Mass	Mass	1200kg
	Initial speed	0 m/s
	Initial way	0 m
Tire Friction	Breakaway friction force	12 N
	Coulomb friction force	12 N
	Viscous friction coefficient	0 Ns/m
Road Load	Breakaway friction force	200 N
	Coulomb friction force	200 N
	Viscous friction coefficient	2000 Ns/m

3.2 Simulate the vehicular system

Your Task:

- 1 Run the simulation for 5 s, leaving all other simulation parameters as their defaults. Plot the vehicle speed, motor speed, motor angular position, and force transmitted in a single PLECS Scope. What are the values of the final translational speed of the vehicle, angular speed of the motor, and force? (3.7 mph, 60 rad/s, and 3571 N).
- 2 Change the motor torque to 1 Nm and re-run the simulation. What are the new values of the final translational speed of the vehicle, angular speed of the motor, motor angular position, and force? (0 mph, 0 rad/s, 0.022 rad, and 35.7 N). Save the completed trace by clicking the “Hold current trace” button on the Scope and label it “Run 1”.
- 3 Change the “Total backlash” of the Gearbox to 0.006 rad and re-run the simulation. What are the latest values of the final translational speed of the vehicle, angular speed of the motor, motor angular position, and force? (0 mph, 0 rad/s, 0.025 rad, and 35.7 N). Save the trace and label it “Run 2”.

- 4 Compare the final motor position and initial transmitted force for Runs 1 and 2. Why is the motor position different? Why is there a delay in the force transmitted? What are the maximum motor speeds for Runs 1 and 2 (0.83 rad/s, 0.96 rad/s, respectively)? Why are these different?



At this stage, your model should be the same as the reference model, `mechanical_domain_2.plecs`.

4 Advanced Motor System Model

4.1 Practical electric drive system

The simple model of the motor uses an ideal torque source to produce the constant torque demand of the driver on the rotating driveshaft. In reality, the motor is an electromechanical device driven by an electrical circuit. The torque demanded by the driver is transformed into a modulation strategy for a three-phase inverter system. The Electric Drive subsystem in the file, `electric_drive.plecs`, represents a typical electromechanical system. It consists of a three-phase inverter, DC voltage source, and a permanent-magnet synchronous machine (PMSM). A Field Oriented Control (FOC) algorithm has been implemented to produce the torque demanded by the driver.



Your Task:

- 1 Replace the Controlled Torque Source and Inertia blocks, representing the simplified motor model, with the Electric Drive block provided in the `electric_drive.plecs` file. Connect the driver torque demand to the “T*” terminal. Change the torque demand back to 100 Nm.
- 2 Use a Signal Multiplexer block to combine the driveline shaft angular speed and angular position signals and feed them into the Electric Drive block’s terminal marked “m”.
- 3 Connect the driveline shaft to the “s” terminal of the Electric Drive block.
- 4 Multiplex the driver torque demand signal and the torque produced by the PMSM and plot the combined signal using a second scope.
- 5 Under menu **Simulation + Simulation Parameters... + Initializations**, load the `mechanical_domain_init` file into the model, which contains the parameter values for the components comprising the full electric drive. This is done by simply calling the file name in the command window as follows: `mechanical_domain_init;`
- 6 Rerun the simulation. Observe the effects of the electrical circuit on the torque produced by the PMSM.

4.2 Sensor error

In a hardware system, sensors introduce additional error to the measured currents and voltages. The measured three-phase stator currents are used to implement the switching signals for the inverter under the FOC control algorithm for the PMSM. A scaling error in the measurement of the phase currents would result in a torque ripple at twice the fundamental frequency of the phase currents. This effect can be realized in the Electric Drive subsystem by introducing an error to the measured currents through the Sensor Error block. The error results in the following measured currents:

$$i_{a, \text{measured}} = (1 + e/100) \cdot i_{a, \text{real}} \quad (5)$$

$$i_{b, \text{measured}} = (1 - e/100) \cdot i_{b, \text{real}} \quad (6)$$

$$i_{c, \text{measured}} = i_{c, \text{real}} \quad (7)$$

where e is the percentage measurement error set in the Sensor Error block.



Your Task:

- 1 Introduce a scaling error of 5% by double-clicking on the Sensor Error block.
- 2 Run the simulation and measure the low frequency torque ripple that is introduced into the system. Measure the fundamental frequency of the phase currents in the Electric Drive subsystem. Confirm the torque ripple has a frequency that is twice that of the fundamental frequency of the phase current by using the data cursors in the scope inside the Electric Drive subsystem.



At this stage, your model should be the same as the reference model, `mechanical_domain_3.plecs`.

5 Conclusion

This exercise has demonstrated a step-by-step approach for creating a simple translational vehicle model using basic Translational components from the Mechanical component library. Additionally, a model of the rotating driveline was developed using components from the Rotational block library. Initially, an ideal torque source was used to apply the torque demanded by the driver on the vehicular system. A more realistic electromechanical system was then provided to observe the sensor error on the torque ripple. An extension to this exercise could be to design an additional controller to actively damp the speed oscillation experienced by the driveline during startup.

The PLECS Mechanical Domain is a useful tool to help engineers observe effects due to interactions between the mechanical and electrical systems. For example, imperfection of the torque-control will have an impact on the load speed and can, in some instances, excite natural resonances. By the same token, mechanical non-idealities, especially of the non-linear kind, will have an impact on the electric system and associated controls. A designer can also investigate the effect of failure modes (of electrical and controls nature).

With complete system-level models, the Mechanical Domain allows for simulating power electronics over their complete use-cycles (for example, a vehicle drive-cycle), which is especially interesting for thermal modeling and reliability analysis. Finally, for complex motion control systems, one can design and test very advanced control algorithms by being able to model the complete mechanical system. This includes servo drives and traction control applications.