

SimDriveline Software

Gažo Martin · Elektrotechnika, MATLAB/Comsol, Študentské práce

28.07.2010



This paper deals with the SimDriveline toolbox, which is used to simulate the mechanical components, primarily used in the automotive industry. Toolbox includes blocks, which can simulate the transmission, clutch, diesel or gasoline engine, and many other mechanical components. The article describes the individual blocks from toolbox and for illustration has been described a simple experiment with two speed transmission and brake.

1. Introduction

SimDriveline software is a block diagram modeling environment for the engineering design and simulation of drivelines, or idealized powertrain systems. A driveline propels a vehicle or craft by transferring its engine torque and rotational power into vehicle kinetic energy and translational momentum. The vehicle moves through or on a medium (air, water, ground) which propels it by reaction forces and which acts as a load on the engine.


Drivelines consist of bodies spinning around fixed axes and subject to Newton's laws of motion. The bodies can revolve about one axis, multiple parallel axes, or multiple nonparallel axes. Simple and complex gears constrain the bodies to revolve together and transfer torque up and down the driveline axes. Locking and unlocking clutches switch the driveline from one gear set to another. Gears and clutches make up transmissions. With SimDriveline software, you represent a driveline with a connected block diagram, like other Simulink models, and you can group blocks into hierarchical subsystems.

You can initiate and maintain rotational motion in a driveline with actuators while measuring, via sensors, the motions of driveline elements and the torques acting on them. You can return sensor signals to the driveline via actuators, forming feedback loops and the basis for controls. The SimDriveline libraries offer blocks to represent rotating bodies; gear constraints among bodies; dynamic elements such as spring-damper forces, rotational stops, and clutches; transmissions; and sensors and actuators. You can also analyze linearized versions of your SimDriveline models and generate code from them.

Driveline Simulation and Physical Modeling: SimDriveline software is based on the Simscape environment, the platform product for the Simulink Physical Modeling

family, encompassing the modeling and design of systems according to basic physical principles. Simscape software runs within the Simulink environment and interfaces seamlessly with the rest of Simulink and with MATLAB. Unlike other Simulink blocks, which represent mathematical operations or operate on signals, Simscape blocks represent physical components or relationships directly.

2. Run SimDriveline

SimDriveline software is organized into a set of libraries of closely related blocks. This section shows you how to open these SimDriveline block libraries and explains the nature of each library. There are several ways to open the SimDriveline block library. You can access the blocks through the Simulink Library Browser. Open the browser by clicking the Simulink button . Expand the Simscape entry, then the SimDriveline subentry, in the contents tree.

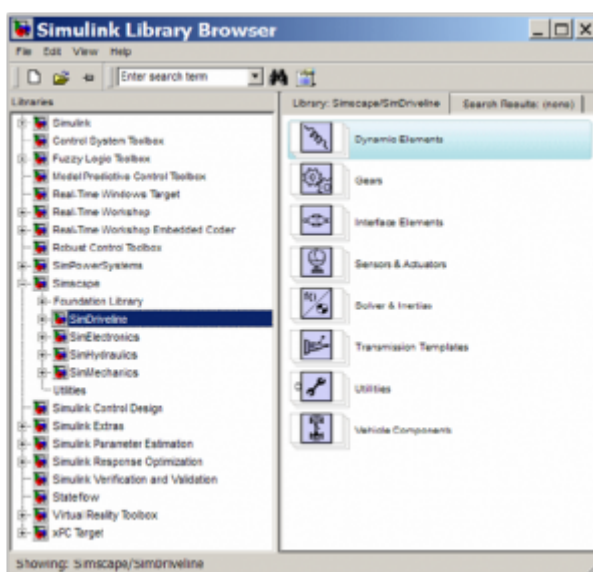


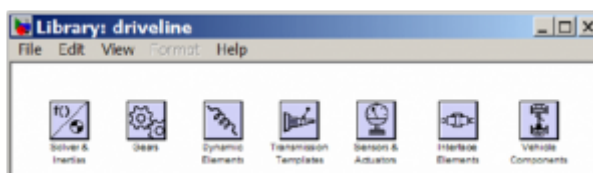
Figure 1 Overview of SimDriveline blocks [1]

You can also access the blocks directly inside the SimDriveline library in several ways:

- In the Simulink Library Browser, right-click the SimDriveline subentry under Simscape and select Open SimDriveline Library. The library appears.
- Click the Start button in the lower-left corner of your MATLAB desktop. In the pop-up menu, select Simulink, then SimDriveline, then Block Library.
- Enter `drivelib` at the MATLAB command line.

3. SimDriveline library

Once you perform one of these steps, the SimDriveline library opens. This library displays seven top-level block groups. You can expand each library by double-clicking its icon.



The next section summarizes the blocks of each library and their use. For explanations of individual blocks, consult the SimDriveline “Block Reference” reference. The SimDriveline block library is organized into separate libraries, each with a different type of driveline block.

Solver a Inertias



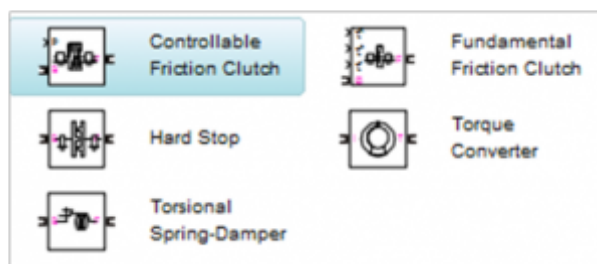
The Solvers & Inertias library provides the Inertia block, which represents a rotating body specified by its moment of inertia, the fundamental unit of driveline modeling. It also contains the Housing block, which represents an immobile rotational ground. Finally, the library contains the Driveline Environment block, which configures the driveline settings of a SimDriveline block diagram, and the Shared Environment block, which allows you to connect two driveline block diagrams in a nonphysical way so that they share the same driveline environment settings.

Gears



The Gears library contains blocks that represent simple and complex gears, driveline elements that couple distinct driveline axes and constrain their relative motions. The Gear blocks range from simple two-wheel gear couplings with fixed and variable gear ratios, to complex multiwheel and multiaxis gears such as planetary and differential gears.

Dynamic Elements



The Dynamic Elements library contains blocks that model such critical drivetrain components as clutches, torque converters, damped springs, and stops. Dynamic elements generate internal driveline torques. The blocks of this library serve as

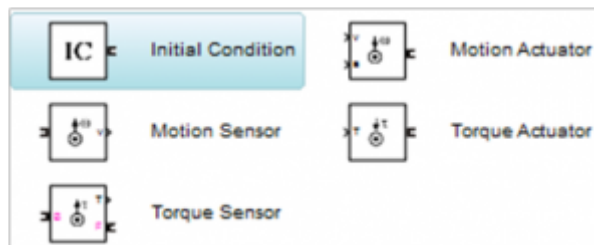
suggestions for developing variant or entirely new models to simulate the same components. Look under the block mask, break the block's library link before modifying it, and create your own version.

Transmission Templates



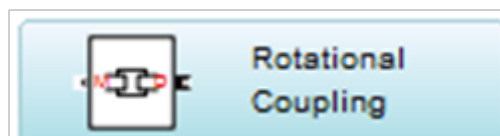
The Transmission Templates are a set of predesigned transmission examples constructed from gears, clutches, and inertias. You can copy and use these examples in your drivetrain models. Transmission templates copied into your model are not linked to the block library. You can modify and rebuild these template copies at will.

Sensors & Actuators



The Sensors & Actuators library provides blocks for sensing and initiating the motions of driveline axes and applying and sensing torques along those axes.

Interface Elements



The Interface Elements library enables connections between SimDriveline driveshaft connection lines and Simscape mechanical rotational motion.

Vehicle Components






Vehicle Components library contains blocks that represent components of a full vehicle beyond the drivetrain itself. It includes models of engines, wheeled vehicles, and tires in contact with the ground. The blocks of this library serve as suggestions for developing variant or entirely new models to simulate the same components. Look under the block mask, break the block's library link before modifying it, and create

your own version.

Essential Steps to Building a Driveline Model

The key rules for connecting driveline blocks to each other and the dual roles of driveline connection lines is transferring torque and enforcing angular velocity constraints. You should review these rules before building and running the tutorial models of this chapter.

- Driveline blocks, in general, feature both driveline connector ports  and regular Simulink inports and outports >. You connect connector ports to one another and Simulink ports to one another. But you cannot connect a driveline port to a Simulink port.
- The driveline connection lines interconnecting driveline connector ports  represent driveline axes and enforce physical relationships. Unlike Simulink lines, they do not represent signals or mathematical operations, and they have no inherent directionality.
- A driveline connection line represents an idealized massless and perfectly rigid spinning shaft. A driveline connection line between two ports enforces the constraint that the two driveline components so connected rotate at the same angular velocity. The connection line also transfers any torque applied to a driveline component at one end to the component at the other end.
- You can branch driveline connection lines. You must connect the end of any branch of a driveline connection line to a driveline connector port .
- Branching a driveline connection line modifies the physical constraints that it represents. All driveline components connected to the ends of a set of branched lines rotate at the same angular velocity. The torque transferred along the input driveline axis is split up among the branches. How the torque is split depends on the dynamical details of the system that you are modeling.
- Driveline connection lines satisfying the angular velocity constraint must have the same initial angular velocities.

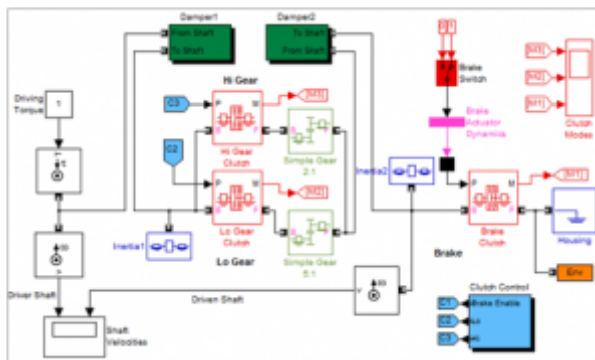
4. Simulation experiments

Two rotating bodies connected with three clutches to switchable dual gears and a brake. Using the Gear and Brake switches to change from high to low gear or to break the motion. Experiment consist of 2 speed gear with transmission rate 5:1 and 2:1 and switchable break. In this simulation we use simple table logic (Table 1) to lock and unlock appropriate clutch.

Table 1 Switching logic table

Transmission State	Clutch1 State	Clutch2 State	Clutch3 State
Neutral/Braked	D/L	Disengaged	Disengaged
Low Gear	Disengaged	Locked	Disengaged
High Gear	Disengaged	Disengaged	Locked

This transmission model couples the gears in a simple way, with each gear and the brake associated with its own respective clutch. Coupling one gear requires engaging and locking its corresponding clutch, while ensuring that the other two clutches are disengaged. Switching on the brake requires disengaging the two gear clutches and locking the brake clutch.



Running the Model, Switching Gears, and Braking [3]

1. After we have built up a model, we have simulated a several actions. First its initial transmission state was low gear. The driven shaft spins at one-fifth the rate of the driver shaft.
2. We have changed the Gear Switch from Low to High, and observe how the driven shaft velocity increases in the Shaft Velocities scope. The driven-to-driver ratio was now one-half. (The driver shaft velocity decreases slightly, because it experiences the damping torque on the driven shaft differently depending on which gear is engaged.)
3. Change the Gear Switch back to Low; we have observed that the driven shaft again spins more slowly. At the same time, while we switched the gears back and forth, the clutches, as shown in the Clutch Modes scope, switch from Lo Gear Clutch being locked, to the Hi Gear Clutch being locked, and back. When one is locked, the other is unlocked.
4. Then we have enabled the brake by changing the Neutral Switch to the upper position. The two gears clutches unlocked and disengaged. The driven shaft, subject to very light damping, now slows gradually. The Brake Clutch remains unengaged.
5. By turning the Brake Switch to on, we can switch the Brake Clutch to the locked mode and bring the driven shaft to an immediate and complete stop. The driver shaft continues to spin at 10 radians/second.

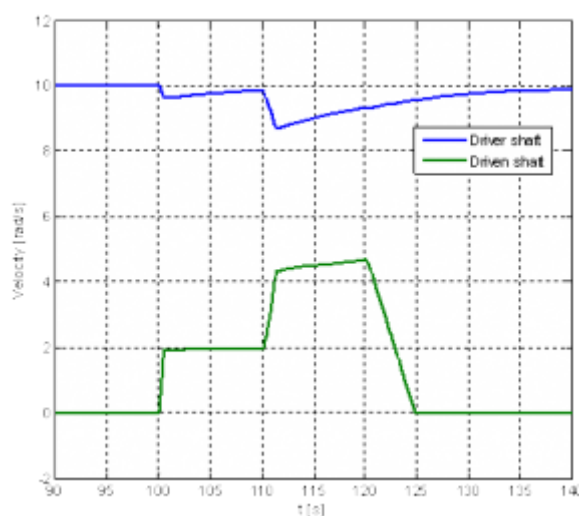


Figure 2 Driver and driven shaft angular velocity

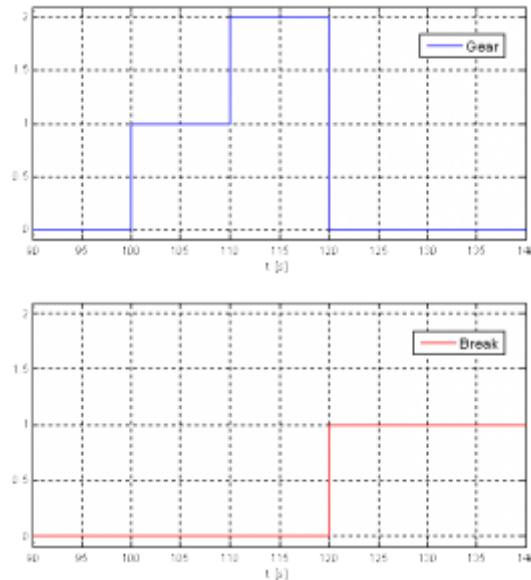


Figure 3 Gear and break switching diagram

6 Conclusion

This toolbox is designed for simulation and modeling of mechanical components primarily use for the automotive industry. In Toolbox there are modeled various components such as gearboxes, couplings, clutches, sensors, engines, wheels and various external environments. These components were used to create simple two speed transmission model, which was transferred to a series of experiments. The simulation schema and the results of experiments are shown in the figures and graphs in the article.

References

1. <http://www.mathworks.com/access/helpdesk/help/toolbox/physmod/drive/>
2. http://www.kxcad.net/cae_MATLAB/toolbox/physmod/drive/bqbdaga-1.html
3. <http://www.mathworks.de/products/simdrive/demos.html>

Coauthor of this paper is Stanislav Triaška, Institute of Control and Industrial Informatics, Slovak University of Technology, Faculty of Electrical Engineering and Information Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia
