

Dual-Clutch Transmission Model Reduction Function

Romain Gillot* Alessandro Picarelli* Mike Dempsey*

* Claytex Services Ltd. Edmund House, Rugby Road, Leamington Spa, CV32 6EL
{romain.gillot, alessandro.picarelli, mike.dempsey}@claytex.com

Abstract

This paper introduces a gearset model reduction function. It is an automated process that works for any type of gear set. The focus is on a dual-clutch transmission (DCT). Reducing the transmission from a multibody model to a table-based loss and inertia 1D rotational mechanics model leads up to a 70% decrease in simulation time. This performance improvement allows engineers to run a detailed physics derived model over the NEDC urban section in real-time or faster.

Many customers have expressed the need for an automated model reduction tool. This paper aims at describing the methodology to create one.

This function will be included in the next release of the VeSyMA – Powertrain library.

Keywords: model reduction, dual-clutch transmission, gear set

1 Introduction

It is an extended version of the function that was introduced in a previous paper (Gillot R., 2017). The functionalities and the process will be explained.

Whenever a full vehicle model is required to be run, simulation time becomes of prime importance. Some compromises can be made in terms of model detail in areas of the model that are not directly the subject of the study. However, the results the simplified subsystems produce still need to be close enough to the ones of the detailed subsystems in order to provide correct interactions amongst the components of interest. The gearbox is one of the most computationally expensive subsystems in a vehicle model. Bearing models with friction enabled, gear pairs with mesh and mesh loss models, shift mechanisms can slow down the model simulation.

This paper shows a method to reduce a gearset model in a quick and automated manner. Essentially, the gear set is run several times with different speed and torque inputs ranging from zero to a maximum defined by the user. This is repeated for every gear. The results of this series of experiments are then collected and mapped and the new reduced model is automatically generated and parameterised.

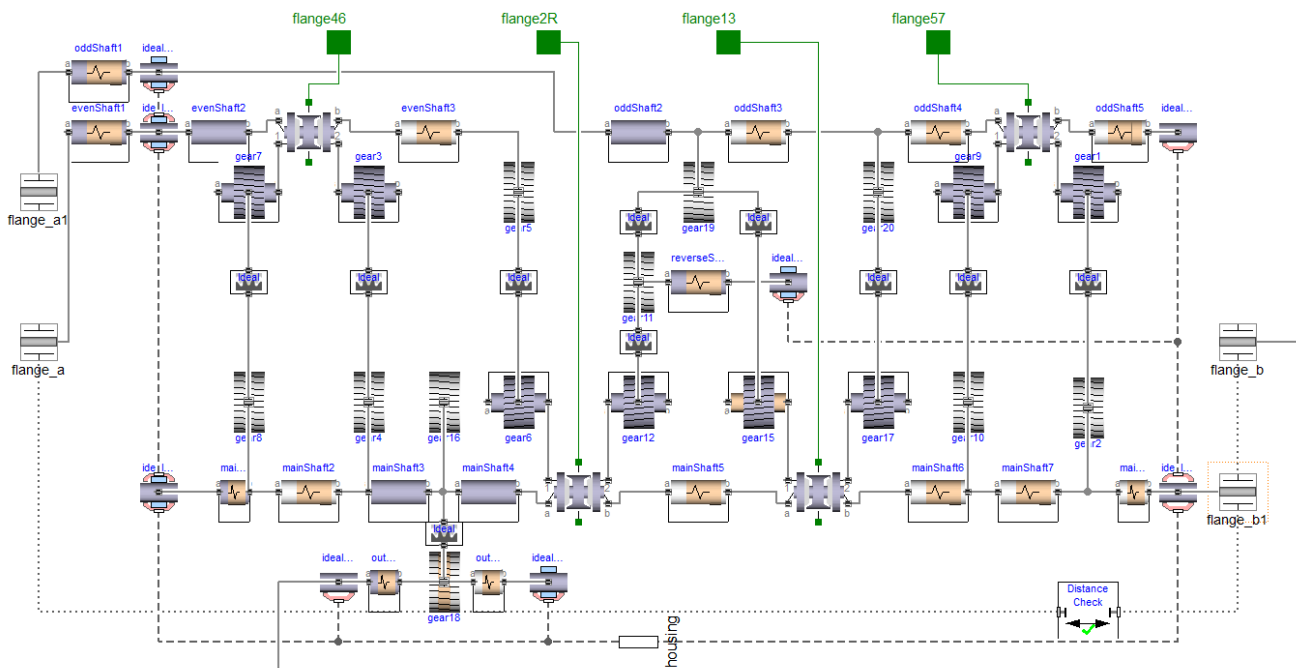


Figure 1. Detailed dual-clutch transmission gear set.

2 Presentation of the model

The 7-speed dual-clutch transmission gearset to be reduced is modelled using components from the VeSyMA – Powertrain library from Claytex. The model interface (i.e. the translational and Rotation3D flanges) is compatible with the Modelica Standard library components.

Friction can be added in the bearings and in the gear mesh models. Several friction models are available for the bearings (Coulomb, Elastic, Hydrodynamic, etc.) and for the gear mesh (Temperature dependent, Torque dependent, etc.).

As this model is detailed, it requires substantial parameterization. The gears need a value for diameter and number of teeth as well as a mesh loss model, the shafts need mass, length, inertia, compliance (optional), the bearings contain detailed friction models.

Parameterizing the detailed gearset in depth to achieve high fidelity is a time-consuming process. We do not want to lose all this valuable information when using the reduced model.

However, it is not immediately obvious what the layout and parameterization of a reduced version of the model above (Figure 1.) should be. We cannot simply use the same parameterization as for the detailed model since the models are different in structure.

3 The function

3.1 The function interface

There are not many inputs required from the user in order to be able to run the function.

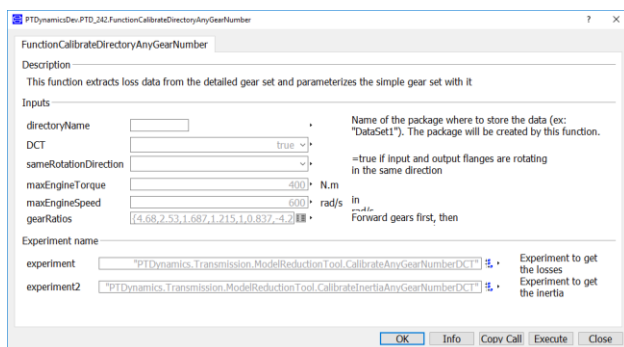


Figure 2. Function user interface.

The user needs to specify:

- *directoryName*: name of the package where the models created by the function (see 3.3.) will be saved
- *DCT*: a Boolean to choose if the transmission that will be reduced is a DCT or not, if not the reduction process is similar, but the model interface is slightly different (only one input flange for example)

- *sameRotationDirection*: a Boolean to specify if the input and output shafts of the detailed model are rotating in the same direction
- *maxEngineTorque*: sets the test upper torque boundary
- *maxEngineSpeed*: sets the test upper speed boundary
- *gearRatios*: transmission gear ratios

Then the correct experiments need to be selected for collecting both the inertia and the losses for generating the reduced model.

3.2 The internal process

This section describes the inner workings of the function

The whole idea is to run the gear set over all its operating points and to collect the results to then map them into a table. The process is repeated for every gear. The final step is to interpolate for speed and torque.

The data collected is: Input and Output shafts inertia, Parasitic Driving and Driven losses and Efficiency Driving and Driven losses.

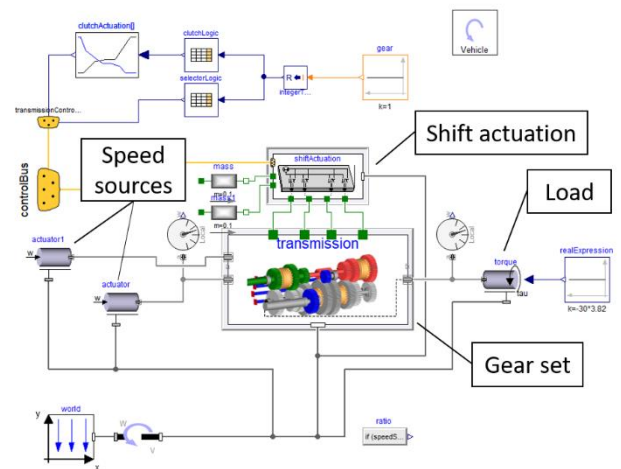


Figure 3. Gear set test rig.

The speed is ramped up to a maximum set by the user which corresponds to the maximum engine operating speed. We repeat this test several times, varying the load every time up to a value specified by the user that corresponds to the maximum engine operating torque.

To make the process more generic and quicker, the function runs the gear set in all the gears at once (including reverse) using arrays of components. This means that the experiment above (Figure 3.) is dragged and dropped into a new model and is of dimension “number of gears”. This means that the function can be used for gear sets with any gear number.

The load-dependent and speed-dependent losses are calculated at the input flanges, the same goes for the inertia. This is true for all the gears except the neutral gear for which we calculate the losses and inertias for the input and output shafts separately since there is no significant physical connection between them.

The last step is to create the reduced model and to parameterise it. This is done using the `print()` function (`Modelica.Utilities.Streams.print`). The simple gear set model (see Figure 4) already exists prior to running the function, it will only be extended during the reduction process to generate a parameterised reduced model.

It should be noted that any gear set model can be reduced using this function, no matter what its inner structure or parameterisation is as long as it uses flanges with bearings as connectors.

3.3 The function output

This is the model that the function will extend from and parameterize.

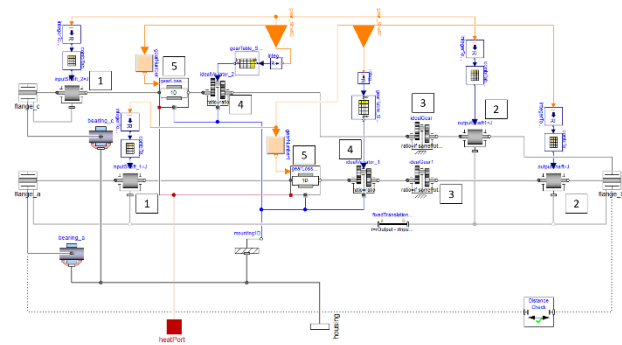


Figure 4. Reduced gear set (1. Odd and even input shafts inertia, 2. Odd and even output shafts inertia, 3. Ratio accounting for rotation direction (-1 or 1), 4. Variable ratio for odd and even shafts, 5. Losses for odd and even shafts).

This is a 1D rotational mechanics gear set model. The interface makes use of 3D flanges for the sake of compatibility with the other vehicle subsystems, but all the rotating components use 1D rotational flanges.

A variable ratio is applied between the input and output shafts. The model uses lumped variable inertias that have a different value for each gear pair selected. The total gear set inertia is lumped at `flange_a` and `flange_c` (the input flanges) except when in neutral, in which case the inertia is accounted for as an input shaft and an output shaft inertia. The losses are also lumped in a table in two places: on the odd and on the even shaft and are interpolated for angular velocity, torque and gear number.

Once the function has successfully run, it opens the model in Figure 4 - after it has been parameterized - along with a set of data records where the data is stored. One data record will be created for each gear that contained the torque-dependent and speed-dependent

losses for both the driving and driven conditions. The inertias will be stored in an additional data record.

Row	1	2	3	4	5	6	7
1	0	0	15.7454	31.7454	47.7454	63.7454	7
2	1.90349	0	0.691249	0.691249	0.691249	0.691249	0.6
3	56.736	0	0.989641	0.989641	0.989641	0.989641	0.5
4	111.569	0	0.994732	0.994732	0.994732	0.994732	0.5
5	166.401	0	0.996468	0.996468	0.996468	0.996468	0.5
6	221.234	0	0.997343	0.997343	0.997343	0.997343	0.5
7	276.066	0	0.997871	0.997871	0.997871	0.997871	0.5
8	283.191	0	1	1	1	1	1

Figure 5. Data record with the collected losses (highlight on the driving efficiency losses).

It takes about 10 to 15 min to run the function.

4 Results

To demonstrate the benefits in using the model reduction function, we ran a full vehicle model over the NEDC Urban drive cycle (see Figure 7), first with a detailed dual-clutch transmission gear set and then with its reduced equivalent.

The solver used is Ccode – variable order with a tolerance of $1e-6$. The number of intervals is 50000 for a simulation time of 800s.

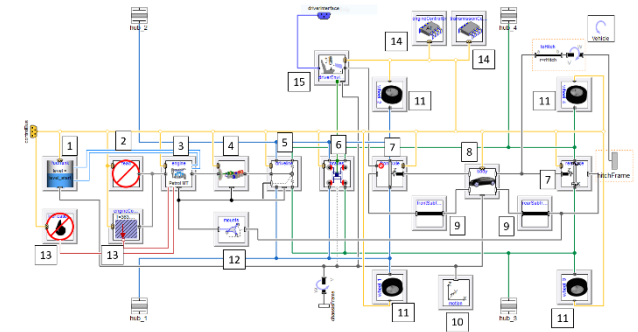


Figure 6. Detailed vehicle model (1. Fuel tank, 2. FEAD, 3. Engine, 4. Transmission, 5. Driveline, 6. Brakes, 7. Front and rear suspensions, 8. Body, 9. Front and rear subframes, 10. Motion block, 11. Wheels, 12. Engine mounts, 13. Cooling and lubrication systems, 14. ECU and PCM, 15. Driver environment).

The vehicle model above (Figure 7) has compliance in many subsystems. The engine mounts, the suspensions and the driveshafts are linearly compliant. Speed and torque dependent losses have been applied to the transmission. Such a level of detail allows to study the vibrations from the engine down the driveline but is computationally very demanding.

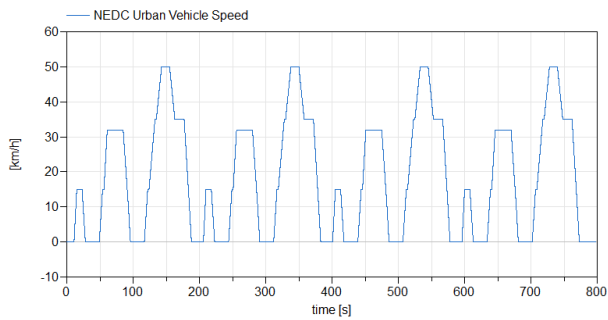


Figure 7. NEDC target vehicle speed.

All the tests described in this paper were performed on a desktop computer with the following processor: Intel® Core™ i5-7600K CPU @ 3.8Ghz. However, the results for simulation speed have been normalised to highlight the gain in performance when reducing the model rather than the overall performance itself. The percentage of speed improvement should be roughly similar on any machine whereas the actual simulation times would be highly dependent on the processor specifications.

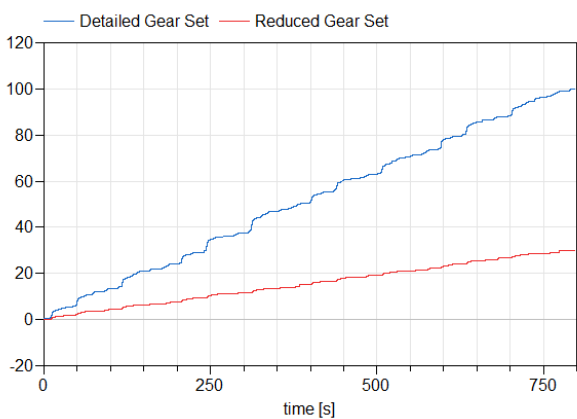


Figure 8. Percentage of CPU time improvement.

The plot above (Figure 8) shows a 70% reduction in simulation time. The vertical axis is CPU (i.e. the time taken by the computer to perform the calculation) and the horizontal axis is the time in the simulation (i.e. the duration of the drive cycle).

The results in terms of accuracy are now discussed. A closed loop driver model is used in this model in order to follow the longitudinal speed profile showed in Figure 7.

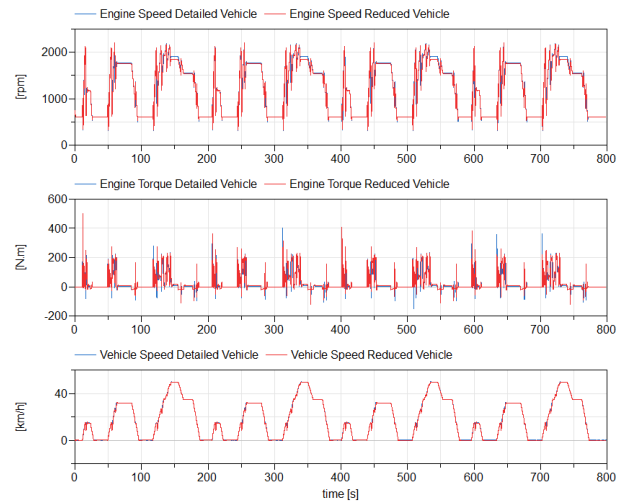


Figure 9. Engine speed and torque and vehicle speed for the detailed and reduced vehicles.

The results are matching well even if there are some small inaccuracies at times, especially visible in the torque plot. The torque map in the engine and losses tables used in the reduced transmission are rather coarse. The increment in the engine speed input of the engine map is 500 rpm and it only uses three different values for throttle opening. The transmission losses maps increase the speed input by 150 rpm and 50 Nm each time. Having coarse tables will lead to inaccuracies during interpolation. Moreover, since the simple engine model used in this paper feeds back the engine speed signal as an input to its torque source, it creates an even bigger error. The last explanation for this error is that the backlash is not taken into account in the reduced model. This means that the torque will be transmitted directly from the input to the output of the gear set with no delay other than related to the compliances in the driveline shafts and mountings. This generates a slight phase shift in the torque curve. Despite these imprecisions, the error remains small (it is investigated further in the following paragraphs).

In terms of the results directly related to the transmission:

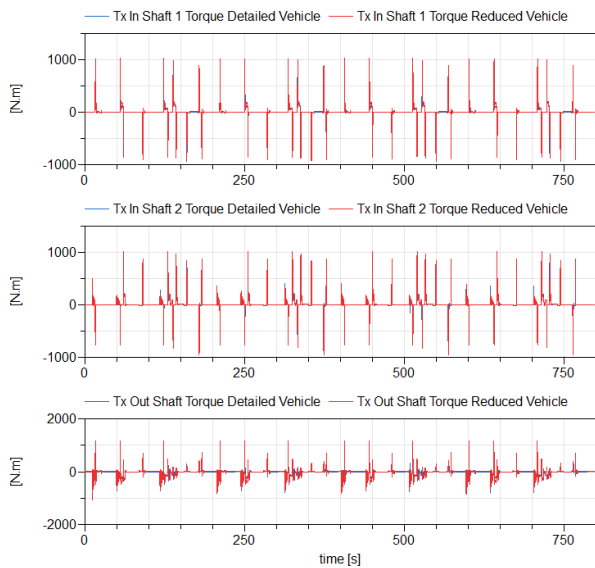


Figure 10. Comparison of the following variables for the detailed and reduced vehicles: transmission input shaft 1 (odd gears) torque (top plot), transmission input shaft 2 (even gears) torque (middle plot) and transmission output shaft torque (bottom plot).

Since the driver controls the vehicle speed, the angular velocity at the input and output flanges of the transmission matches very well the results of the detailed model. It is thus more interesting to analyse the torque curves.

In Figure 10 we can observe a slight discrepancy in the results. When the transmission input shaft speed changes sign, we can see first a slight phase shift of engine torque. The reduced model not considering backlash will thus start to transmit torque a few moments before the detailed model.

Another error happens during gear shift. The reduced gear set model does not use synchronisers to ensure high shift quality. This will result in transients that are accurate enough to allow this reduced model to be for this type of studies (gear shift, transient operation). Of course, one should keep in mind that such a simplified transmission model is not suited for studies which focus on this subsystem for driveability and shift quality. The level of detail in each vehicle subsystem needs to be adjusted depending on the area of interest.

5 Conclusion

A function has been created that provides an easy routine to reduce a dual-clutch transmission gear set model. The performance improvement is significant and the reduction process is quick (it takes less than 10 minutes on my machine). The accuracy of the results is overall satisfying even if some particular operating conditions lead to some considerable error. Attention has to be given to what type of study is to be carried before considering using this reduced model. An

improved accuracy is however to be sought for in the future to broaden the range of tests than be done using this model.

It could be extended to other subsystems in the future.

Given all the gearset models in the array used to generate data for the reduced gearset are independent from each other, we could thus take the advantage of the multicore capability and run each one of them on a separate core. This needs to be investigated as part of future work.

References

- Gallagher Stephen. et al. Model-based Real-time Systems Engineering, Loughborough, England, Powertrain Modelling and Control Conference, 2016
- Dempsey Mike. et al Coordinated automotive libraries for vehicle system modelling, Vienna, Austria, Proceedings of the 5th International Modelica Conference, 2006
- Dempsey Mike. et al. Investigating the Multibody Dynamics of the Complete Powertrain System, Como, Italy, Proceedings of the 7th Modelica Conference, 2009
- Gillot Romain, Picarelli Alessandro, Dempsey Mike. Model Reduction Techniques Applied to a Physical Vehicle Model for HiL Testing, Prague, Czech Republic, Proceedings of the 12th Modelica Conference, 2017