

# Anticipatory Shifting – Optimization of a Transmission Control Unit for an Automatic Transmission through Advanced Driver Assistance Systems

Salim Chaker<sup>1</sup> Michael Folie<sup>2</sup> Christian Kehrer<sup>1</sup> Frank Huber<sup>2</sup>

<sup>1</sup> ITI GmbH Dresden, Germany, {chaker, kehrer}@itisim.com

<sup>2</sup> IPG Automotive GmbH, München, Germany {michael.folie, frank.huber}@ipg.de

## Abstract

By integrating system simulation with vehicle dynamics into real-time environments, it is possible to simulate the physically correct behavior of vehicle components and also adjust it to the required operating strategies depending on external factors. Multi-physics system simulation for realistic representations of powertrains and their behavior combined with dynamic driving simulation allows for optimizations of the transmission control unit for an automatic transmission by employing advanced driver assistance systems for an increased efficiency through anticipatory shifting.

Realistic load cases that are based on measured data help optimize fuel consumption and driveline dynamics with respect to the control algorithms by using variation calculations with variable transmission parameters. This works also the other way around when control algorithms are validated and optimized quickly and free of risk as part of rapid prototyping.

*Keywords:* Automatic Transmission, Advanced Driver Assistance Systems, CarMaker, SimulationX, Modelica, FMI

## 1 Introduction

During the development of modern transmissions in the automotive sector – as for any technical system – there is a growing complexity caused by more and more features and functionalities. In order to get reliable test results for making a profound decision, it is inevitable to integrate such interacting functionalities into a virtual prototype, which has already become common practice. Especially advanced driver assistance systems monitoring the surrounding area through a number of sensors to generate an environmental model for situation-based interpretations contribute to the growing relevance of simulation solutions.

The more and more diverse interactions between the many sub-systems spanning across various physical domains, such as mechanics, electronics, hydraulics or control engineering, can be modeled in an object-oriented way including their dynamic effects. With a higher efficiency in mind, these multi-physics models are used for various scenarios during the development and analysis of new transmissions, such as hybridization, cylinder deactivation or the integration

of advanced driver assistance systems. The optimal utilization of available potentials can only be achieved by combining means of transmission tuning and optimizing operating strategies. As a consequence, advanced driver assistance systems are more and more developed with the focus not only on safety aspects, but also on the reduction of fuel consumption and on anticipatory driving. Many functionalities and applications are already in use or are being developed. Tools which can perform certain tasks of engineers in a virtual world including software development, applications and testing are becoming increasingly indispensable. The objective of the work presented in this paper is to develop a virtual verification environment for gearbox control algorithm with focus on fuel consumption.

## 2 Modeling an 8-speed automatic transmission

### 2.1 Modeling paradigm

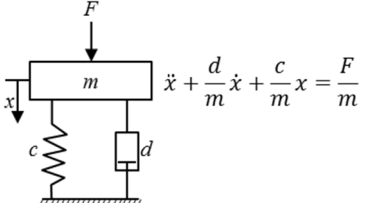
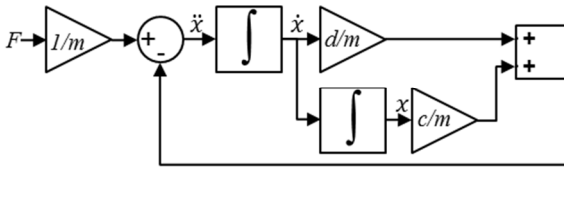
For the representation of technical systems, there are a number of modeling options available. A principle categorization may be based on the interactions within a model. Signal-based modeling (e.g. Matlab/Simulink, ETAS ASCET) shows the functional correlations based on the principles of control engineering by using blocks with inputs and outputs. This includes the direction of effect and thus the solution direction, i.e. causal modeling. As real physical systems barely show a direction of effect, it is often necessary to solve the differential equation systems that describe the technical system. Especially complex systems with multiple components which are connected on a differential-algebraic level may push single-oriented models towards their limits.

Acausal approaches are not subject to such limitations so that also huge systems can be modeled intuitively. Instead of inputs and outputs, there are connectors functioning as interfaces for flux variables (e.g. current, inertia, force) and potential variables (e.g. voltage, angle, distance). The overall equation system is a result from the inner system equation of the components and the equations from the connections (potential variable equation, flux variable balance). This declaratively described equation system is symbolically transformed and solved in advance

allowing for a very time-efficient simulation. The most prominent example for object-oriented, acausal modeling is the Modelica language (P. Fritzson, 2004).

The major features of both signal-oriented and object-oriented acausal modeling are listed below (tab. 1) for the respective representative Matlab/Simulink and Modelica/SimulationX (ITI GmbH 2013). For

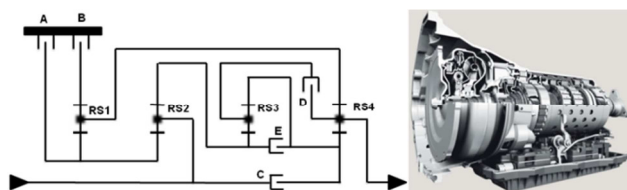
modeling the transmission's control loop, object-oriented modeling is the preferred option, while signal-oriented modeling in Matlab/Simulink is the tool of choice for vehicle parts related to control engineering (engine and transmission control units, operating strategies).

Acausal, object-oriented modeling (Modelica/SimulationX)	Causal, signal-oriented modeling (Matlab/Simulink)
	
<ul style="list-style-type: none"> <li>• Direct implementation of differential-algebraic equations to describe technical systems</li> <li>• Free solution direction (acausal modeling with differential-algebraic equations)</li> <li>• Symbolic analysis and forward solution of the equation systems (order reduction), then numeric integration (→ minimizing the computational effort)</li> <li>• Easy creation of complex systems by connecting them through interfaces similar to real systems</li> <li>• Models with identical structure to real systems (high model readability)</li> <li>• Class structure and inheritance (high re-usability, extendibility)</li> </ul>	<ul style="list-style-type: none"> <li>• The implementation of differential-algebraic equations requires reformulating/rearranging them in order to define the solution direction (or re-modeling an inverse model if needs be)</li> <li>• Numeric solution block by block, no analytic summary or forward solution</li> <li>• Time-consuming creation of complex systems through manual solution of the involved differential equation systems</li> <li>• High level of abstraction of the real physical system (bad model readability)</li> <li>• No inheritance or class structure (bad re-usability and extendibility)</li> </ul>

**Table 1:** Comparison of modeling paradigms

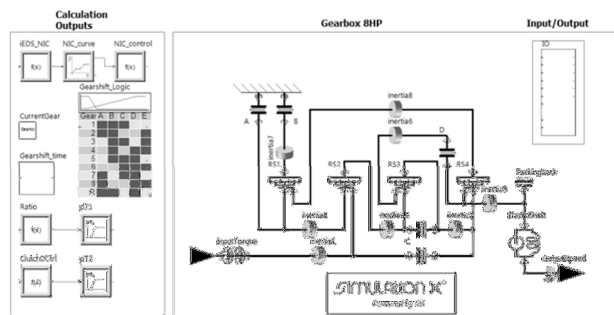
## 2.2 Model of the 8HP transmission

The ZF 8HP is an 8-speed automatic transmission by ZF Friedrichshafen AG. The transmission consists of four planetary gear sets and five shift elements (three multiple plate clutches and two brakes). Figure 1 shows the interaction between the many different elements involved. Gears are changed by opening and closing the shift elements. Depending on the selected gear, the shift elements are open or closed in order to achieve a certain transmission ratio by adapting the distribution of force through the planetary gears. This is handled by a transmission control unit which was modeled in Matlab/Simulink to take advantage of the benefits of signal-oriented modeling.



**Figure 1.** Transmission diagram of the ZF 8HP automatic transmission

SimulationX is a continuous CAE solution for acausal/object-oriented modeling, simulation and analysis of physical effects. The tool comprises ready-to-use model libraries for 1D and 2D mechanics, 3D multibody systems, power transmission, hydraulics, pneumatics, thermodynamics, electronics, electric drives, magnetics and control engineering. The 8HP transmission including its hydraulic actuator structure was modeled in SimulationX with elements from the 1D rotatory mechanics library and with signal blocks (see Fig. 2). The model received also an interface for inputs and outputs matching those of the gearbox modeled on the open integration and test platform CarMaker in order to allow for a seamless integration.



**Figure 2.** SimulationX model of ZF's 8HP automatic transmission.

The desired gear or sequence of gears for dynamic gear changes can be assigned to the mechatronics module `schaltLogikVerz` (see left side of Figure 2) that implements the controls of the brakes (A, B) and the clutches (C, D, E) of the 8HP transmission. Clutch and brakes were modeled with elastic friction points including signals for (de-)activating the torsional flux between the drive components through a controller.

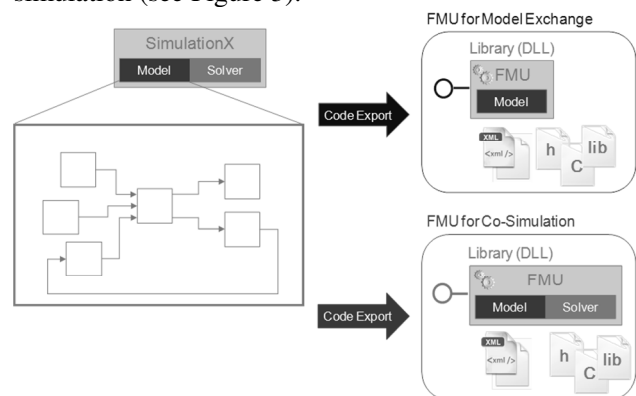
### 2.3 Model exchange via FMI

Due to the high interoperability between SimulationX and CarMaker, it was an easy choice to go for a dual approach combining the benefits of both tools. In order to implement such a simulation concept, model exchange capabilities or a coupled simulation between the chosen tools is inevitable. This has the following advantages:

- Platform independent availability of component models for OEMs
- Black box models for IP protection during model exchange between companies
- Option to use models from different modeling environments with certain characteristics and benefits that can improve the model quality and simulation efficiency
- Re-usability of models regardless of the modeling environment (FMI (Functional Mock-up Interface) standard [3]) as well as easy testing and validation of models
- Realization of distributed simulations for coupled models from different domains and for the optimal utilization of the available resources

This principle was further promoted with the FMI standard specified in the MODELISAR project [3]. This interface standard allows for transfers functionalities of an entire SimulationX model in a FMU (Functional Mock-up Unit). Other simulation environments (e.g. CarMaker) can then instantiate the generated FMU and can access the unit's functionalities through the FMI interface (FMI Association, 2010). There are two types of the FMI

standard: FMI for model exchange and FMI for co-simulation (see Figure 3).



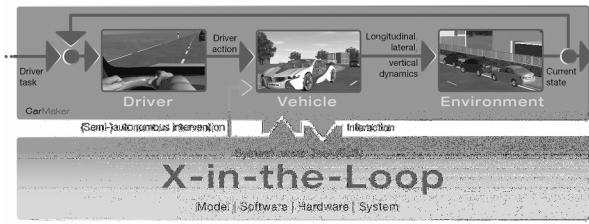
**Figure 3.** FMI code export for model exchange and for co-simulation.

While for FMI for model exchange, the model's behavior is exported as C code and entirely integrated and simulated in the target simulation environment, FMI for co-simulation involves two or more simulation tools that are coupled with each other. The model's behavior is exported as C code together with the solver. The generated FMU for co-simulation runs as a slave in the target simulation environment and exchanges the computed results at discrete communication points with the master environment. Between these points, the individual solvers compute the sub-systems independently from each other.

### 3 Integration and test platform CarMaker

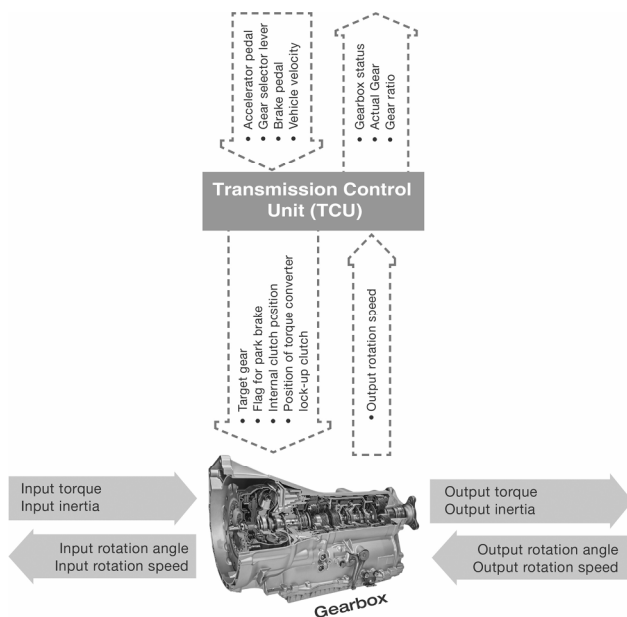
CarMaker and the related products TruckMaker and MotorcycleMaker offer plausible models for driving physics and driving environment as well as virtual drivers capable of performing relevant maneuvers of ordinary drivers, but also of test drivers. As a test platform, CarMaker provides also maneuver characterizations based on real driving behavior. Also complex open and closed loop test scenarios can be realized.

Besides vehicle and driver models, CarMaker also includes a complete environment simulation with streets (curves, traffic signs, traffic lights etc.), moving traffic with near field sensors and the integration with digital maps (e.g. ADAS-RP by HERE, Google Earth). This creates a very realistic representation of the desired test environment. CarMaker also allows for a consistent implementation of the X-in-the-loop approach (XiL) (Figure 4). The XiL method permits early integration into the complete vehicle and comprehensive validation of relevant system components, be it as model, software or hardware (B. Schick, 2013).



**Figure 4.** XiL permits early verification of systems. CarMaker provides the necessary interfaces to integrate all relevant components and systems into the virtual vehicle.

As an open integration and test platform, CarMaker offers an interface architecture tailored to the needs of the automotive industry (Figure 5). At the click of a button, models, software components and actual vehicle components are integrated into so-called digital prototypes – from the single component to integrated systems. If necessary, individual powertrain components, the chassis, assistance or control systems, for instance, but also concepts of operations or display functions can be integrated (B. Schick, 2012). The virtual integration is the basis for the analysis of occurring influences of the tested components and functions on the vehicle's behavior. Flaws in the development process can thus be identified early on (S.-A. Schneider, 2012).



**Figure 5.** Realistic interfaces with gearbox components are necessary for an easy and continuous integration. Additional inputs and outputs can be included.

The digital prototype systems can be verified as a whole during virtual test driving. Such test drives can be repeated and modified if needed, while the test results are valid up to the physical limits. Virtual test driving and realistic driving scenarios permit evaluations of new developments across the entire V model (H. Palm, 2013).

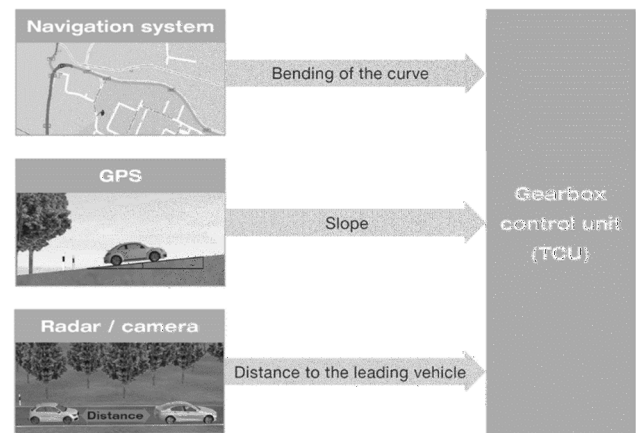
#### 4 Driver assistance systems in the powertrain and the gearbox

Today's automatic transmissions must meet various requirements, such as weight, design space and shift comfort, but also aspects for reduced fuel consumption. As mentioned earlier, assistance systems can register and interpret environmental information and pass it on as data to systems and components.

The following section focuses primarily on shift strategies for which the mechanics module schaltLogikVerz received additional software modules. It demonstrates how these functionalities can be ensured and optimized in virtual test drive scenarios. This involved the transmission model needs to be integrated as a SimulationX FMU with the controller logic for the gearbox as a Simulink FMU in CarMaker. The software for the shift strategies has access to additional environmental information (Figure 6):

- Curvature of street curves
- Slope of the street
- Distance to the vehicle in front

CarMaker obtains the information about curvature and slope from the navigation system via the standardized ADASIS protocol. The distance to the vehicle in front is determined through a radar or camera model. All data captured through sensors are passed on to the transmission's software and can be processed.



**Figure 6.** Information captured through sensors and passed on to the transmission's software.

The aforementioned information can now be used for the development of such features as:

- Avoiding gear changes in a curve or finishing a gear change before the next curve,
- Engaging idle mode downhill to benefit from coasting,
- Using recuperation during deceleration or downhill to keep distance to the vehicle in front or to charge the battery.

The last two requirements are complementary. On the one hand, the kinetic energy should be used more effectively and the friction effects in the transmission should be reduced to a minimum, while on the other hand the battery should be charged through the wheel movements. A side-effect of this is that the engine slows down the vehicle which makes using additional external braking redundant at times, e.g. to keep the distance to the vehicle in front. Variation calculation can now help find the best gear change strategies for various slopes and distances to the vehicle in front. The integration of the transmission's plant model and software can be combined into various MiL, SiL, HiL, especially test bench, scenarios:

- MiL involves the transmission model and the software to be integrated in CarMaker as an FMU.
- SiL replaces the transmission's software with a production code which can be integrated in CarMaker as C code or as an AUTOSAR FMU.
- In the next step, the transmission software is replaced with a real control unit as part of a HiL setup.
- At the end, also the gearbox FMU is replaced with a real gearbox on a test bench.

CarMaker remains the integral tool throughout all development stages. The test cases created in the MiL phase as well as the vehicle model and the parameterization can now be used in all of the shown phases. The generated data reflects a high degree of reproducibility and is also suitable for testing and improving the transmission's software. The transmission test bench allows for certain physical effects to be included in the transmission model, which in turn contributes to a higher quality of the transmission control software.

It is then possible to also include the driver in further analyses. He can choose from either manual mode for overtaking maneuvers, for example, or switch to cruise or eco mode. These options can also be combined in several variations of different tracks and distances to the vehicle in front. That results in a large number of tests which can only be conducted in a reasonable fashion through MiL or SiL scenarios due to performance aspects. HiL and the test bench are only employed to test load cases for current, pressure and momentum, but also for certain limits of the defined spaces in order to minimize the number of tests.

Finally, driver input as well as vehicle and transmission data can be captured during the real test drive. The driver input and the velocity are used as input signals for the simulation in all phases in order to be able to compare the results from the various simulations with the actual test drive. Only a profound

model calibration can make a solid basis for variation calculations that are true to life.

## 5 Summary and outlook

The combination of shorter development cycles, an ever-growing model complexity, the level of detail and variation as well as the vast amount of reliable analyses to be conducted early on in the development process poses a huge challenge to an engineer's daily routine nowadays. Modeling, simulating and processing input data as well as the representation of the generated results in a digestible way require much effort. The object-oriented, acausal modeling approach meets the needs of the user, and the intelligent optimization algorithm delivers an efficient simulation. Due to the advantage of this modeling paradigm, more and more heterogeneous domains are combined with each other and simulated in just one model. The simulation performance with heterogeneous models reaches its technical limits. The implementation of a distributed simulation, however, through co-simulation as described above, for example, offer an excellent opportunity to master such challenges.

The separate integration of transmission and software offers several benefits – e.g. the reduction of the number of pre-defined variants or a quick way of virtually testing new concepts. Comparing real and simulated driving behavior creates a valid basis for virtual test driving. As a consequence, it is now not only possible to test and optimize safety relevant functions early on and efficiently in a closed toolchain, but also aspects of driving comfort and fuel consumption.

The described concepts also allow for fuel saving display functions in vehicles with manual transmissions. Such could include for example „Foot off the gas“ or „Shift to idle mode“. Other fuel saving features could be integrated with driver assistance sensors. Traffic sign and traffic light detection could be used to increase efficiency for coasting and recuperation phases. These ideas will be part of future developments. They will also focus on other aspects of virtual testing and the optimization of operating strategies for hybrid and non-hybrid vehicles, such as Car2X or car2car. After all, it is those test drives that are difficult or even impossible to conduct in real life that are the driving factor in the field of virtual validation.

## References

- P. Fritzson. Principles of Object-Oriented Modeling and Simulation with Modelica No 2.1, 2004.
- ITI GmbH. SimulationX User Manual, Dresden 2013.
- FMI Association. Functional Mock-up Interface, MODELISAR (070006), 2010.
- B. Schick. Mission V-Process Enhancement by Integrated Vehicle Performance Evaluation within an Entire X-in-the-Loop Process, *Keynote SIAT ARAI*, 2013.

- B. Schick, V. Leonhard u. S. Lange. Vorausschauendes Energiemanagement im virtuellen Fahrversuch. *ATZ* No 4: 328 – 333, 2012.
- S.-A. Schneider, B. Schick, H. Palm. Virtualization, Integration, and Simulation in the Context of Vehicle Systems Engineering, *embedded world 2012*, 2012.
- H. Palm, J. Holzmann, A.-S. Schneider, H.-M. Kogeler. Die Zukunft im Fahrzeugentwurf, Systems-Engineering-Basierte Optimierung, *ATZ* No 6: 512– 517, 2013.