

A new Modelica Electric and Hybrid Power Trains library

Massimo Ceraolo¹

¹DESTEC Department, University of Pisa, Italy, massimo.ceraolo@unipi.it

Abstract

This paper describes a new library proposed for simulation of electric and drive vehicle power trains.

Since is a “first approach” library, it does not make usage of the Vehicle Interface Library. It does not overlap with that library, except for minor parts, since the proposed models of electric drives and battery are much more detailed than the simple examples available in the Vehicle Interfaces Library.

This library is fully compatible with both Dymola 2015 and OpenModelica 1.9.2. It is available under the Modelica License 2, and presented at the 11th Modelica International Conference

Keywords: Power Train, Electric drive, Model, Synchronous machine, Asynchronous machine, map-based model, Electric Vehicle, Hybrid Vehicle, Power-split device.

1 Introduction

This paper shows a new small modelica library that is devoted to simulation of vehicular electric power trains.

There are some important reasons to have a specific library for this purpose.

Electric propulsion of vehicles is very important nowadays, since it involves both pure electric vehicles and electric-hybrid vehicles. Its simulation requires somewhat specific models: in fact, detailed models of electrical machines require the variables to follow their sinusoidal variation, that can have frequencies of hundreds or thousands of hertz. Detailed simulation of power converters is even more demanding, since commutation frequencies are easily up to 50 kHz, and therefore time steps must be as small as a few microseconds. This is overkill for vehicular simulations that typically simulate trips lasting several minutes. For instance, the well-known standard vehicle simulation cycle, the NEDC, lasts 1200 s. As a consequence of this, to simulate vehicular propulsion, averaged models must be used, that are sufficiently precise and yet not too demanding in terms of simulation resources.

The library therefore supplies models of electric drives that are a compromise between detail and resource requirements that is adequate for electric vehicles. In addition it supplies other support models that are very important in vehicles, i.e. a battery model and a model for the vehicle drag force.

The library contains four main folders: *MapBased*, *ElectricDrives*, *SupportModels*, *Icons*, *FullVehicles*.

ElectricDrives contains models for synchronous and asynchronous electric drives and simple examples showing their behaviour and how they compare to more detailed MSL models

MapBased. For even more simplified simulations, map-based models of electric drives are satisfactory. Here some models of this kind are proposed, that are tested and then used in some of the full vehicles examples. This folder contains also two ECUs (Electronic Control Units) that are used to control the map-based full vehicle examples.

SupportModels. This folder contains useful models needed, in addition to electric drives, to create full vehicle models: two battery, a drag force and a driver model. The subfolder “Internal” contains models that are internally used to interface mechanical and electrical parts of models, and are not intended for final user usage.

FullVehicles. This folder contains examples of full vehicle power trains built using the supplied models: two electric and to hybrid vehicle models are present.

All the models of this library have been checked and work well with Dymola 2015 and OpenModelica 1.9.2.

The remainder of this paper illustrates the supplied models as well as the full vehicles examples

2 Electric Drives folder

This folder contains models of electric drives based on asynchronous and synchronous machines.

2.1 Asynchronous machine models (AMDrive and AMDrivePU)

The asynchronous machine models are based on Modelica.Electrical.QuasiStationary library. This way, the only dynamics of the machine and drive come from the rotating parts: indeed electrical time constants are typically much lower than those due to mechanical inertias.

Moreover these models offer the advantage of easy interpretation, since the graphical representations reproduce closely the quasi-static single-phase models of electrical machines commonly found in textbooks.

Although the asynchronous machine models provided here are not adequate to simulate machine behaviour unbalanced currents and heavy transients, they cover very satisfactorily the simulations commonly found in electric and hybrid vehicle power trains.

The core of these models is as depicted in figure 1.

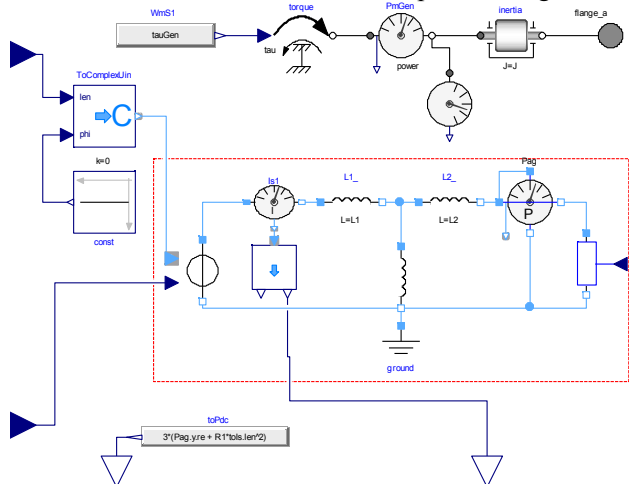


Fig. 1: Representation of the QSAsma model inside the ElectricDrives folder.

The signals entering the model, left in the figure, are the voltage behind stator resistance (rms per phase) and frequency. These are applied to the classical quasi-stationary single-phase asynchronous machine equivalent circuit, where air-gap power and thus torque are evaluated. The generated torque is then applied to the machine inertia (upper part of the picture).

Although it is possible to use this model as is, it was built to be used inside other library models, i.e. AMDrive and its per-unit version AMDrivePU.

The AMDrive arrangement is shown in figure 2.

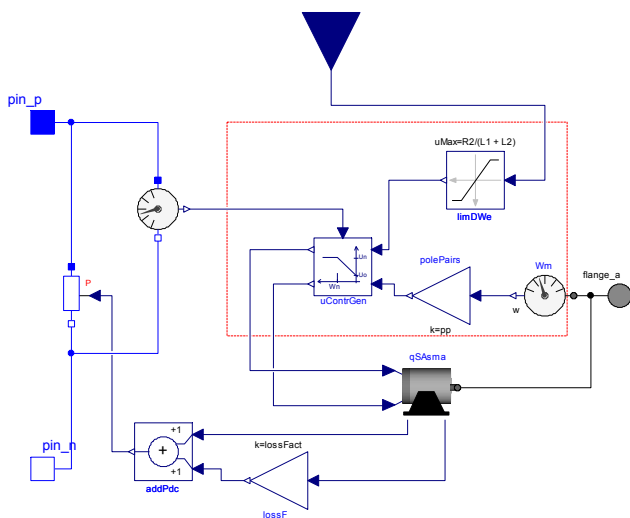


Fig. 2: Internal structure of the proposed WSDrive component.

The previous asma model of figure 1 is used and fed by a control system “uContrGen”, that generates voltage behind stator resistance, according to a classic voltage proportional to frequency rule (Bondea, Nazar, 2006).

Note that the input signal (above in the figure) is the slip frequency dWe , that is intended to be as torque request. In fact, if dWe is not too large, it is nearly-proportional to the asynchronous machine electro-

magnetic torque, as well known from that machine theory. This happens only up to $dWe = R2/(L1+L2)$; that is why the limiter $limDWe$ is introduced in the scheme.

The drive model computes the power absorbed by the drive as the sum, obtained in $addPdc$ model, of $qSAsma$ absorbed power and additional losses created in the $lossF_$ block, proportional to machine stator current, that is often a reasonable estimation of inverter losses this because on-state losses often dominate and are proportional to the AC current. The $qSAsma$ absorbed power is computed in $qSAsma$ model as the sum of mechanically generated power and copper losses in $R1$ and $R2$ resistances. However the user can easily enhance this model including additional losses such as iron and stray losses.

A simple test of the asynchronous machine drive is supplied in the library, ElectricDrives. | TestingModels folder, named TestAMDrive.

In this test the proposed drive performance is compared with a similar drive, but obtained based on the MSL standard model of asynchronous machine. This standard model is fed with voltages created using the $U/f=const$ technique similar to that used inside the proposed $qSAsma$, but with quantities that have the actual variable-frequency sine shape.

Just to have an idea of the expected results, in figure 3 the terminal voltages of the two models are compared to each other. Note that $qSAsma$ model gives the rms value; the figure confirms that its behaviour is that of the aimc terminal voltage peaks divided by $\sqrt{2}$.

Similarly, in figure 4 the produced electromagnetic torques are shown.

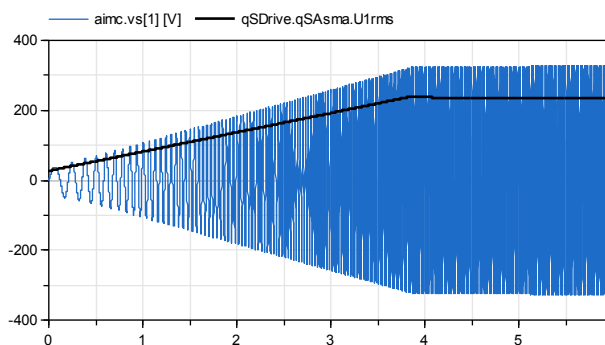


Fig. 3: Comparison of MSL aimc and $qSAsma$ model during a start-up: instantaneous (vs, from MSL), and rms ($U1rms$, from $psDrive$) phase voltage.

A more complex example using the $QsDrive$ model is provided in FullVehicles library folder and described later on in this paper.

A variant of AMDrive called AMDrive PU is also provided. The only difference is that the parameters are expressed in per-unit. This could be useful when the precise machine to be simulated is not known; in this case one could envisage some p.u. values and typically change only, between simulations, the nominal apparent power $Snom$.

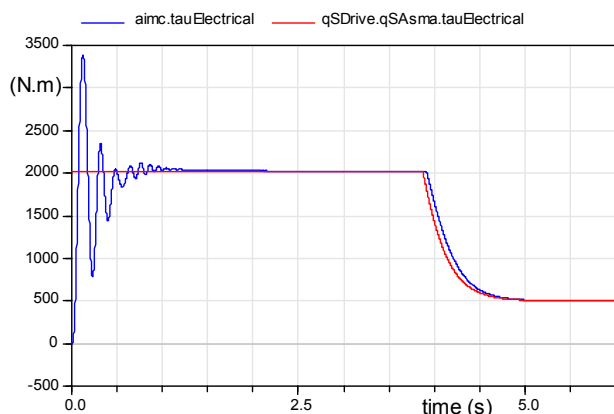


Fig. 4. Comparison of MSL aimc and qSAsma model during a start-up: aimc and qSDrive electromagnetic torques.

2.2 Synchronous, permanent-magnet machine model (PMDrive)

Permanent magnet machines are more and more used on vehicles because of the very advantageous characteristics. Maybe the most important ones are the possibility to have a very large speed range over base speed, and a higher specific power, in comparison to asynchronous machine. The latter feature is mainly due the fact that PMs produce magnetic field without the need of current circulation, differently from the squirrel cage of asynchronous machines. Thus they do not require rotor cooling (Ehsani *et al*, 2006).

The control of PM machines must guarantee that the angle between rotor-generated and stator generated magnetic fields are optimal, i.e. that guarantees the Maximum Torque Per ampere (MTPA) condition. In case of isotropic machine this optimal angle is 90° , while in case of anisotropic one, where the direct-axis reluctance is lower than the quadrature axis one, this optimal value is more than 90 degrees (Schiferl, 1990).

Often instead of the angle between the two fields, the difference between this angle and 90° is considered: this new angle, called gamma-angle, has an optimal value of zero for isotropic machine, while it is larger than zero when $X_q > X_d$. An idea of the optimal gamma-angle trend can be obtained looking at figure 5, in which example combinations of PM flux, and X_d and X_q are considered.

The optimal gamma angle is very easily computed in Modelica: it is just necessary to impose:

$$0 = -\psi_{PM} \sin \gamma + (L_q - L_d) I_s \cos 2\gamma \quad (1)$$

That is just an additional equation to be added to the machine equation set.

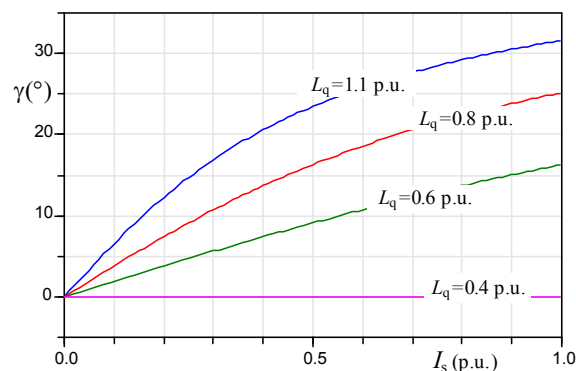


Figure 5. Trend of the optimal gamma angle as a function of stator current for a machine having a magnetic flux $\Psi_{PM}=0.6$ p.u., a direct inductance $L_d=0.34$ p.u., and different values of L_q .

However, as is well known, the optimal gamma can be selected only at low speeds: at very high speeds the flux produced by permanent magnets tends to generate a too high terminal voltage and this tendency must be contrasted by a flux weakening part of the stator current produced field. So the condition determining the gamma angle at these speeds is the one that creates the set voltage at the machine terminals:

$$V_{machine} = V_{set} \quad (2)$$

So equation (1) will be substituted by equation (2) thus retaining the variable-equation balance of the model.

Equations (1) and (2), along which an if-equation to switch between the two, is enclosed in the “AtomicPmsm” model present in the SupportModels folder of the library. This is complemented by a loop that avoids the machine current to overcome a set value in the PmsmAllFluxLimI model, also present in the SupportModels folder. In case of conflicting needs for current, i.e. current is needed to accomplish flux weakening and to produce the requested torque, priority is given to flux weakening.

This way, the usage of PMDrive model, that is rather concise (128 equations) a lot of the PM drive characteristics are provided for:

- automatic selection of optimal gamma angle according to (1) at low speeds
- automatic flux weakening at higher speeds
- automatic switching from optimal angle and flux weakening control
- automatic current limitation
- consideration of machine and inverter losses.

A simple test of the permanent-magnet synchronous machine drive is supplied in the library, ElectricDrives|TestingModels folder, named TestPMDrive.

In this test the proposed drive performance is compared with a similar drive, but obtained based on the MSL standard model of synchronous machine. This standard model is fed with voltages created using pre-

defined Id-Iq pairs, and converting these currents into their time-domain counterpart

Creating a logic that reproduces the optimality obtained in pMDrive just for sake of comparison is too demanding. Therefore in this simulation the values of Id – Iq are simply read in the pMDrive part of the simulation, approximately reproduced by means of two trapezoidal shapes, and then fed into the MSL smpm machine.

Figure 6 shows how the trend of Id (left) and Id (right) are reproduced by trapezoidal sources in the smpm model

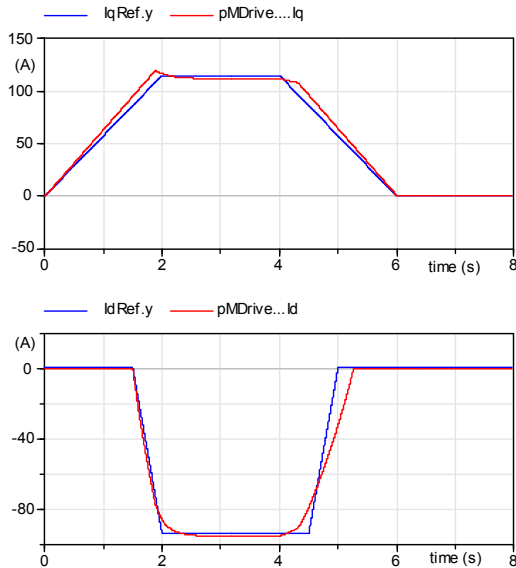


Fig. 6. Trend of Id (left) and Id (right) as reproduced by trapezoidal sources in the smpm model.

Figure 7 shows the voltage on phase 1 as well as the norm of the space-phasor machine voltage that should in principle equal the peak of the sinusoidally varying phase smpm voltages. The red curve shows how the space phasor voltage amplitude $V_{spFF} = \sqrt{V_d^2 + V_q^2}$ could be in case flux weakening would not take place¹.

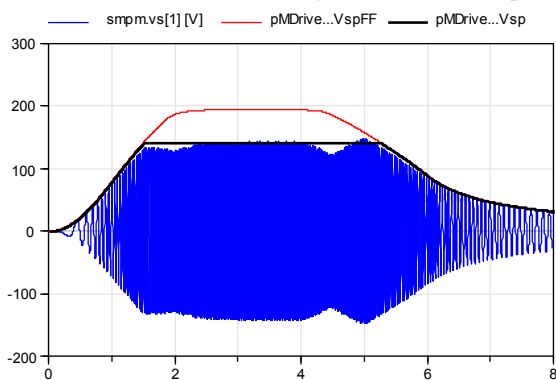


Figure 7. Phase voltage (blue) computed Space Phasor (SP voltage amplitude in absence of flux weakening (red), set SP voltage amplitude (with flux weakening, black).

¹ The two letters “sp” stand for space phasor”, and “FF” stand for “Full-Flux”.

Finally in figure 8 the electromagnetic torque generated in the MSL smpm machine and in the library’s PMDrive are compared.

3 Map-Based folder

In many circumstances the dynamics to be considered in vehicle power train studies are much slower than the faster electric dynamics. In these cases, the only state variables to be considered in electric drive models can be those related with the mechanical inertias of the rotating parts. The rest can be modelled as being algebraic, i.e. with maps containing operating regions and efficiencies.

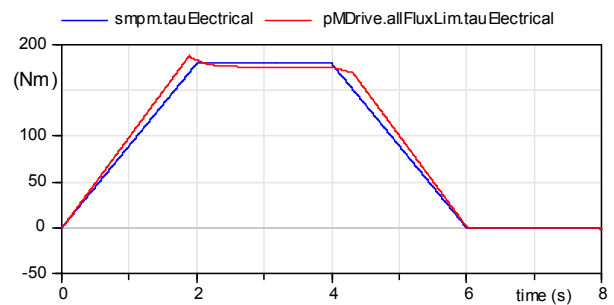


Fig. 8. Comparison of MSL smpm and PMDrive model during a start-up: smpm and PMDrive electromagnetic torques.

To ease simulations in these cases the map-based folder has been provided in the library.

The general arrangement of map-based components can be understood looking at the MBOneFlange model, whose inner architecture is shown in figure 9.

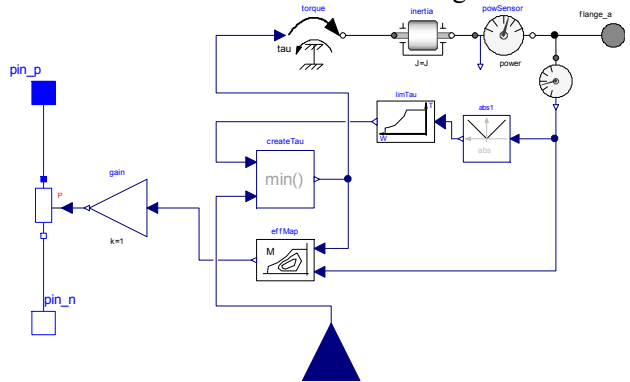


Fig. 9. Internal structure of the MBOneFlange model.

This model models an electric drive, that tries to produce and apply to the inertia the mechanical torque requested from the real input tauRef.

Before applying that torque to the inertia it is verified if the requests is compatible with the drive torque limits as determined by the limTau component, that, as usual in the electric drives, impose a limitation on the delivered torque and delivered power, whatever comes first.

Once the torque is applied to the inertia the applied torque and actual speed determines the operating point of the electric drives. This point is used by the effMap

component that computes the drive losses as a function of the operating point, and requires the absorption from the DC flanges of the total drive power, including losses, by means of a variable resistor, that has as input the power to be drawn.

In a similar way also two –flange electric drive model is internally built.

As regards the ICE model, it follows the same rationale, but instead of efficiency maps it is deemed more natural to use fuel consumption maps The architecture is thus the one shown in figure 10.

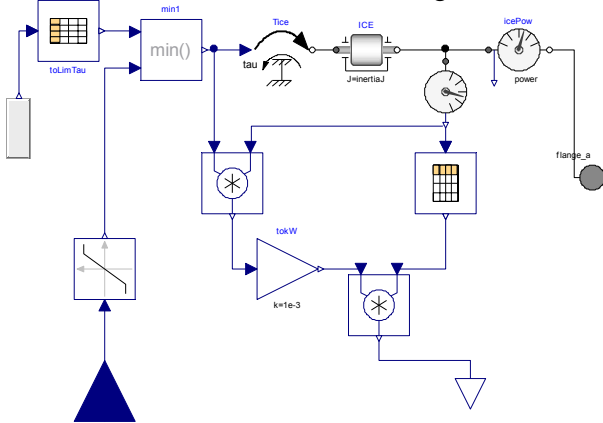


Fig. 10. Internal structure of the MBice model.

The fuel consumption is computed by means of the map “toSpecCons” whose output is in g/kWh, that is multiplied in the block “toG_perHour” times the kW power, to generate the wanted consumption expressed in g/h.

The map-based components come in two flavours: with the set torque being a Real input signal, and with a bus connector.

The components having the bus connector have the advantage that the connectors carry several useful signals, instead of just the set torque. However they have some disadvantages:

- the final user must use the correct names for the bus signals. The usage of connector-based components is recommended, at least at first, either in association with the provided Electronic Control Unit models MBecu1 and MBecu2, or using the converter blocks MBSupport| ToConnIceTauRef and MBSupport| ToConnGenTauRef. Examples of the ECU models usage are provided in FullVehicles folder, while examples of the MBSupport converter blocks are provided in MapBased| TestingModels whose names contain the word “Conn”.
- At most one MBice, one MBOneFlange and one MBTwoFlange components are simultaneously allowed, unless some changes are made on the supplied MB models with bus connectors.

The names that interface with the bus in connector components are those shown in table I. Note that the OneFlangeConn component is called “gen”, while the TwoFlangeConn component is called “mot”. This is a

choice that tends to simplify things, and is consistent with the meaning of the full vehicles models FullVehicles| Psecu1 and FullVehicles|Psecu2.

Table I Names and meaning used as bus signals on the components containing bus connectors.

Name	Sender	Unit	Meaning
iceTauRef	*	Nm	Torque that the ice is requested to deliver
iceW	MBiceConn	rad/s	ICE speed
icePowDel	MBiceConn	W	Power delivered by ICE
genTauRef	MBOneFlangeConn	Nm	Torque that the gen is requested to deliver
genPowDel	MBOneFlangeConn	W	Power that the gen delivers
genTauLim	MBOneFlangeConn	Nm	Maximum torque gen can deliver at the actual speed
motTauRef	MBTwoFlangeConn	Nm	Torque that the mot is requested to deliver
motPowDelA	MBTwoFlangeConn	W	Power that the gen delivers through flange A
motPowDelAB	MBTwoFlangeConn	W	Power that the gen delivers summing flange A and flange B outputs
motTauLim	MBTwoFlangeConn	Nm	Maximum torque gen can deliver at the actual speed

*This is the reference torque that is put on the bus by either ToConnIceTauRef or ToConnGenTauRef or MBecu1 or MBecu2.

All the map-based components are tested in specific testing models.

Here, just as an example, the testMBOneFlange is presented and discussed.

The model is represented in figure 11. It consists on a one-flange drive that is requested to follow a torque profile. The requested torque is larger than that the drive can follow.

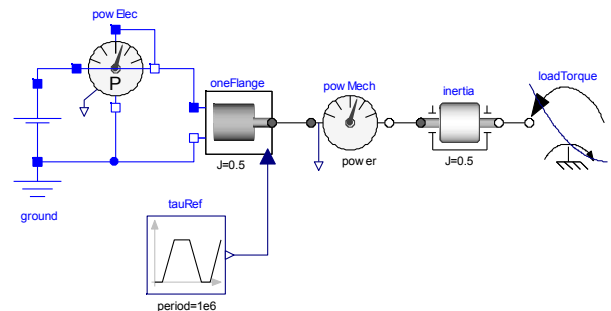


Fig. 11. Test model for the MBOneFlange model.

Some significant plots are shown in figure 12.

It can be noted that:

- during the first 10 seconds the generated torque oneFlange.torque.tau, is 20Nm, as requested from the input. The maximum torque that can be generated is not limited by the power limit (thus state=0)
- between t=10s and 14s the generated torque continues to follow the input signal; but starting from t=10.8s the maximum torque that can be

delivered is limited by the maximum drive power (this is confirmed by the value state=1)

- between t=14 and 18 s, since the drive power has been reached (10 kW), the generated torque is automatically reduced to avoid this limit to be overcome
- between t=18 and t=38 the maximum speed is reached and therefore the generated torque is automatically reduced to avoid this limit to be overcome (state=2)
- above t=38 the torque request is reduced and the drive is again able to deliver this torque.

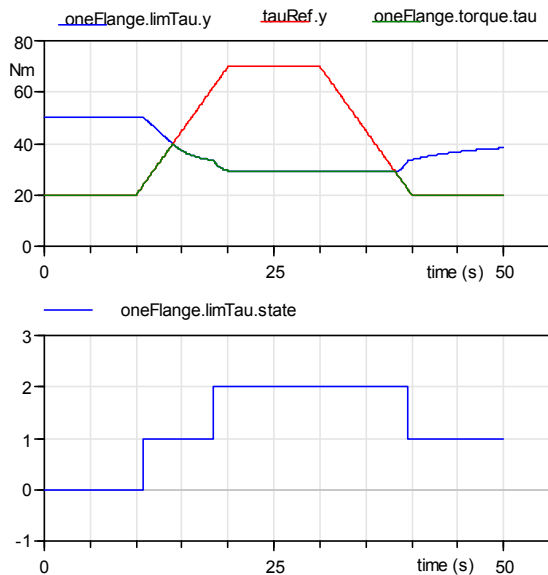


Figure 12. Torques (above) and state (below) of the TestMBOneFlange test model.

Mechanical and electric powers are shown in figure 13. In the central part of the transient, in which speed is constant, the ratio of the two powers represents also the efficiency of the drive in that operating point, that in this case is 85.7%

4 Support models

4.1 PropDriver

Simulation of vehicular power trains normally needs drivers to be simulated.

Indeed in the past also the so-called inverse simulations were common, and used for instance in Advisor software, at least in its royalty-free version distributed by USA’s Department of Energy up to 2003 (EERE 2015). But in recent years direct simulations, typically more realistic and accurate, have become the standard. Direct simulations require the vehicle driver to be simulated along with the vehicle power train.

In the first release EHLibrary, discussed in this paper, a very simple driver model is proposed: the model reads from the hard disk the speed-time profile to be followed, and tries to follow it by means of a purely proportional controller.

Although so simple, it allows very useful simulations to be performed, as demonstrated in the full Vehicle model example provided (section 5).

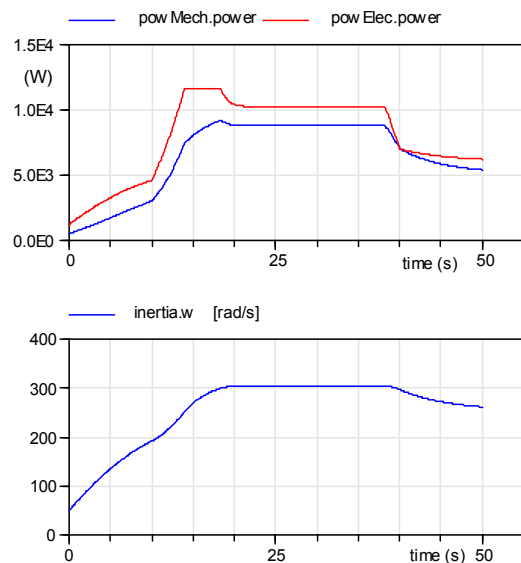


Figure. 13 Mechanical and electrical powers (above) and rotational speed (below) of the TestMBOneFlange test model.

4.2 DragForce

The resistance to movement in vehicles in flat rows is usually expressed by the following well known formula:

$$R = mgf + 0.5 * \rho S C_x V^2 \tag{3}$$

The first term simulates rolling resistances, proportional to vehicle mass m by means of the rolling coefficient f , the second aerodynamic resistance, and therefore is proportional to the air density ρ , front area S , drag coefficient C_x , squared speed.

Indeed formula (3) does not tell all the truth: if we use it in simulations, at zero speed R would be non-zero and the vehicle would start backwards.

To correctly simulate resistance to movement hybrid simulation is needed, e.g.:

$$R = \begin{cases} mgf + 0.5 * \rho S C_x V^2 & \text{if } |V| > 0 \\ F_{est} & \text{if } V = 0 \end{cases}$$

Where F_{est} is the applied external force. The vehicle will start moving whenever F_{est} overcomes mgf .

Fortunately this is easy in Modelica, and the Drag Force component implements this hybrid set of equations. It is directly derived from the MSL component Modelica | Mechanics | Translational | Components | Brake.

4.3 Batt1 and Batt1Conn

A very important piece of hardware for electric and hybrid vehicles is constituted by electrochemical batteries.

The model proposed here derives from the many years of experience of the author in battery modelling, with some useful hints coming from the “BatteryIdealized” model available in the Dassault Systèmes’ SmartElectricDrives library (Dassault Systèmes 2015).

For instance, papers (Ceraolo 2000, Barsali 2002) discuss a whole family of models in which different numbers n of R - C blocks are used to simulate battery dynamics. Although developed for Lead-acid batteries, these models have then proven to be valid for other kinds of batteries, and in particular lithium batteries (Ceraolo 2011) A choice of n corresponds to a given compromise between complexity and precision. In the EHPowerTrain the compromise chosen refers to one R - C block, so that the model can be represented by the equivalent circuit shown in figure 14.

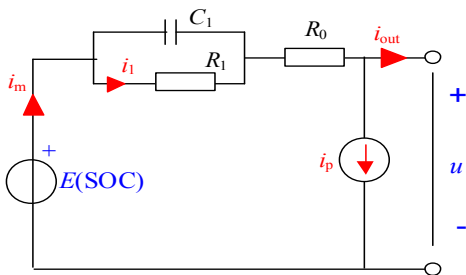


Fig. 14. Battery Equivalent circuit with EMF.

In this model the current that contributes to the charge/discharge process is i_m , while the current flowing in the parasitic branch R_p , i_p is lost. Indeed, the state of charge of the battery can be computed starting from the so called “extracted charge Q_e ” that is the integral of i_m :

$$SOC = 1 - \frac{Q_e}{C_Q} = 1 - \frac{1}{C_Q} \int_0^t i_m(t) dt \quad (4)$$

In (4) it is assumed that when $t=0$ the battery is completely charged, so that Q_e is the charge extracted from the main branch of the electric circuit (the branch in which i_m flows) starting from a fully charged battery.

In general, all the circuit parameters: i_p , R_0 , R_1 , C_1 are function of state of charge and electrolyte temperature; moreover i_p is a non-linear function of the terminal voltage. But for the purpose of EHPowerTrain all these dependences are neglected, except the most important one, i.e. the dependence of E on SOC.

Indeed this dependence in many batteries is nearly linear:

$$E = E_0 + E_1 SOC = E_0 + E_1 \left(1 - \frac{Q_e}{C_Q}\right)$$

Thus in these cases the law relating i_m to E is the same as the one relating current and voltage in a capacitor. In EHPowerTrain linear dependence of E on SOC is assumed, and therefore the circuit of figure 14 is converted into the circuit shown in figure 15.

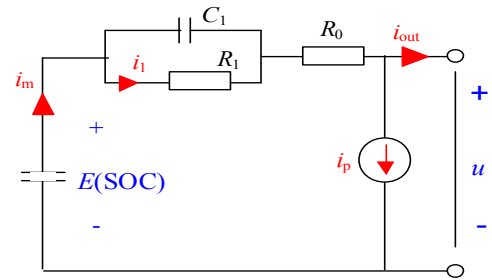


Fig. 15. Battery Equivalent circuit with Capacitor-EMF.

It must be noted that it is expected that the library user knows little about i_p of our model.

Therefore, using a technique that can also be found in SmartElectricDrives Library, i_p is indirectly computed from the full charge/discharge efficiency of the battery. I.e., the user specifies the global constant-current charge/discharge efficiency and from this datum a constant i_p is determined.

Naturally, since i_p cannot be lower than zero and adds loss to the energy loss due to R_0 and R_1 , the user-defined battery efficiency must be not lower than that corresponding to $i_p=0$ condition.

5 Full Vehicles examples

To realistically simulate full vehicles one of the best ways is to use the freely available Vehicle Interfaces library (Modelica Association 2015). However simpler models allow understanding basic things about vehicles more easily. Therefore the EHPowerTrain comes with full vehicle examples that are built from scratch using just modelica and MSL; it is not difficult, however, to include some of the supplied models in the interfaces available in the VehicleInterfaces library, this way taking best of both libraries.

There is one very simplifying assumption that must be noted in the proposed FullVehicles models: the driver outputs a single signal that is a torque reference. When positive it is intended to be traction torque, as could be drawn from the position of the accelerator pedal, while when negative is intended as a brake torque, as could be taken from the brake pedal. This unique torque is sent to the electric or hybrid power trains, that takes care of accelerating or braking. I.e., it is supposed that the power train can perform all the braking actions needed by the speed profile, without additional intervention of mechanical brakes.

These full power train models are stored in the folder named “FullVehicles”. More models are intended to be added in future versions of the library.

In the present version of the library the provided models contain two electric vehicles, one of which (EvAm_bat) based on an asynchronous machine, the other (EvPm_bat) on a permanent-magnet synchronous machine.

The other two (Psecu1 and Psecu2) refer to two hybrid power trains realised with the Power Split

Device component, one of which keeps always on the engine, the other has also some ON/OFF strategy.

Here, for illustration of what these models can do, some results on EvPm_Bat and PSecu2 are proposed.

5.1 EvPm_bat model

This simple vehicle model shows how a PMDrive can be easily and effectively exploited in electric vehicle models. Its appearance is shown in figure 16.

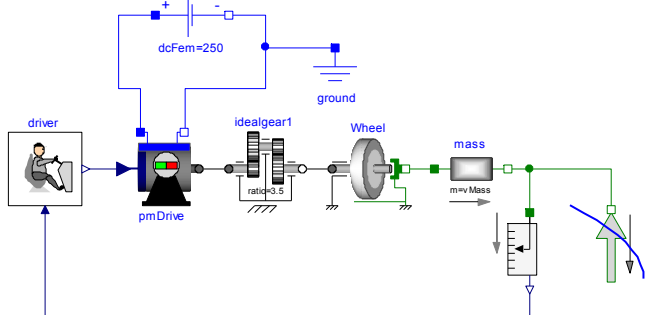


Figure 17: Graphical view of the Evm_bat model.

In this simulation the driver, simulated by means of a simple proportional controller, tends to follow the drive cycle.

The considered drive cycle is the so called "Sort1" standard [UITP 2010]. It is composed by three simple triangles of speed versus time: each of them is composed by three phases: constant acceleration, constant speed, constant deceleration. The maximum speeds of these three triangles are 20 km/h, 30 km/h, 40 km/h respectively.

Figure 18 shows some significant plots.

The proposed simulation refers to a rather small car having a mass of 1300 kg.

In the upper graph the vehicle speed $mass.v$ as long as the wished vehicle speed $driver.from_kmh$. It is seen that this driver is very reactive, and therefore the vehicle speed closely follows the set driving cycle.

In the central plot the generated electromagnetic torque $pmDrive.allFluxlim.tauElectrical$ is shown: the plot indicates that the inertia (acceleration and deceleration) forces dominate. During the constant-speed parts the torque is due only to drag forces, composed of rolling friction and air drag. Since the considered speeds are low, the air drag force is negligible.

Finally, the lower graph shows trend of $I_d(t)$ and $I_q(t)$ that indicate the machine behaviour. Since this machine is anisotropic, torque is determined by I_d and I_q . If an isotropic machine is chosen instead, I_q will determine torque, while I_d will determine terminal voltage; I_d will stay equal to zero at low speeds, to reduce drive current.

Other interesting plots can be drawn from the electrical DC circuit. In the next figure 19, the power delivered by the battery and the State of Charge (SOC) are shown.

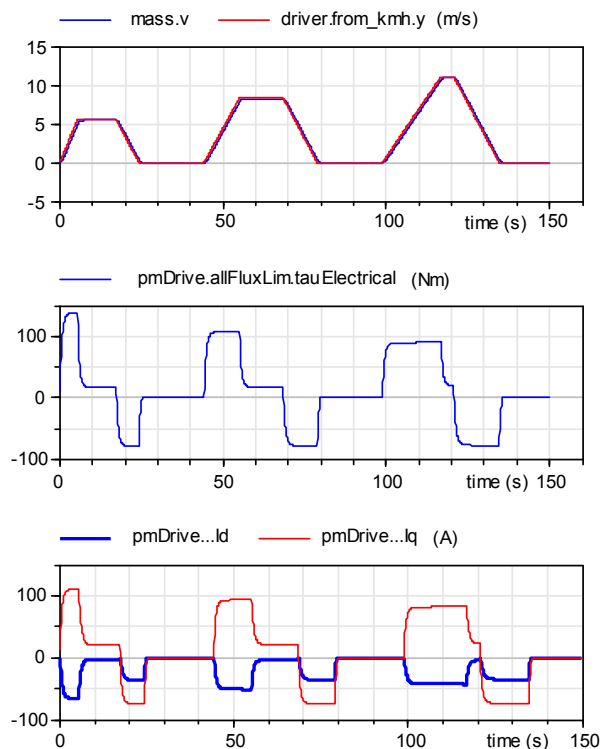


Fig. 18. Some significant plots of EvPm_bat model.

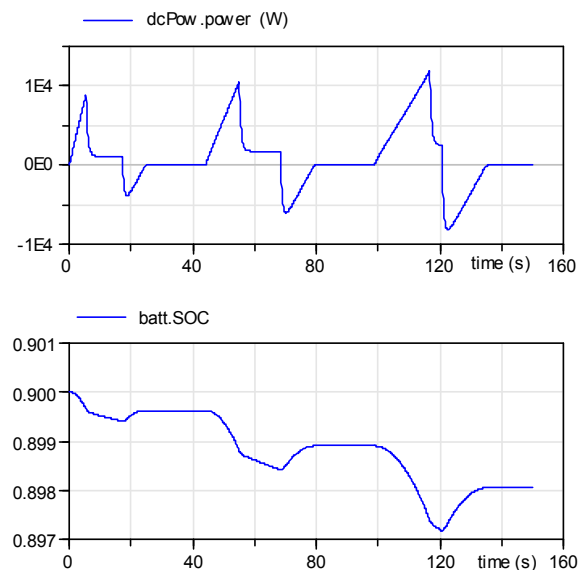


Fig. 19. DC power and SOC of EvPm_bat model.

Note the very small SOC window due to the very short simulation: only three bus stops, for a total of 0.5 km.

The drag force model understands when the vehicle is standstill and switches into locked mode. This can be verified checking as shown in figure 20.

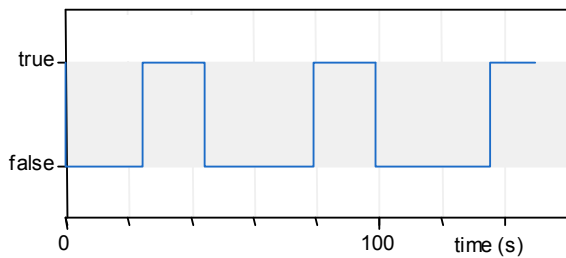


Fig. 20: Value of the boolean variable “dragForce.locked” for the simulation shown in figures 18 and 19.

5.2 PSecu2 model

This model shows a possible behavior of a power train based on a Power Split device (PSD) model. Its appearance is shown in figure 21.

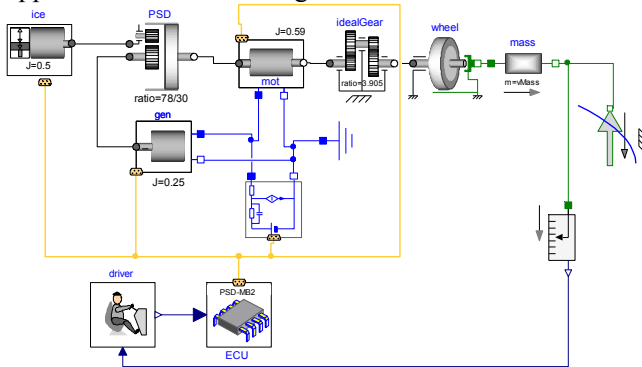


Figure 22: Graphical view of the PSecu2 model.

Moreover it gives the opportunity to show several of the Map-based library models, in the version containing a bus connector. It thus exploits MBOneFlangeConn, MBiceConn, MBTwoFlangeConn components.

The management of all these devices is made in the ECU of the type MBecu2. It tries to satisfy the driver’s torque command in an effective way, considering the ICE consumption fuel map. To do this it has an inner logic that foresees also switching off the ICE at low loads.

The general idea of this logic, that reproduces what is published regarding the first release of Toyota Prius PSD based hybrid, and described in (Toyota, 2003) is as follows:

1. When the ICE is ON it is made operate at its maximum torque, i.e. at the maximum throttle, that for any engine speed corresponds to the lowest specific consumption. Moreover that torque is roughly independent on the ice speed, being always around 90 Nm; this implies that control of the ICE power is obtained just controlling the ICE speed.
2. As a general rule, the ICE speed is chosen to be the value that makes it deliver the load power, measured as an average on the last few minutes of vehicle operation.
3. The one-flange machine, connected to the PSD sun and called “gen”, has as purpose to keep the ICE near its optimal speed.

4. The two-flange machine, connected to the PSD ring through one flange and the final reduction gear through the other, is operated so that the vehicle follows the torque requests from the driver
5. The above logic is modified to keep SOC under control; this control is obtained by action on ICE speed obtained, again, in compliance with the above rule 3, by means of additional action on the gen.
6. Finally, if the engine is delivering for long time too low power, it is temporarily shut down, using a simple ON/OFF technique. Determination of too low ICE power is made by corresponding measure of ICE speed (compare rule 1 above); an hysteresis loop is added to avoid too frequent ON/OFF actions.

When it is decided that switching off the ICE is needed, its reference speed is brought to zero. The given consumption map is such that at zero speed there is also zero consumption. Naturally, although reasonable this is an approximation. In a real case the ICE fuel injection would be brought to zero, and then when also its speed reaches zero some mechanical brake would be activated to allow the PSD to exchange torques with the mechanical objects connected to the other flanges.

In the following figures some results of the simulation proposed in the EHPowerTrain library are proposed.

The considered vehicle is the same as for the EvPm_Bat model. Even the total mass of the two vehicles is taken as equal. However here instead of the Sort1 Cycle, the New European Driving Cycle (NEDC) is used.

Figure 21 shows, from top to bottom, the desired and obtained vehicle speed; the ice rotational speed (when it is brought to zero ICE is set to OFF position and no fuel consumption occurs); the battery SOC that, despite of some fluctuation, shows a general trend to stability due to the specific control loop present in PSecu2.

6 Conclusions

This paper has shown the basic characteristics of a new library, EHPowerTrain, proposed to be presented at the 11th Modelica International Conference.

It is intended for people wanting to simulate electric and hybrid powertrains, with lean models and fast simulations.

The models have thus defined with limited complexity; yet they are able to give interesting results.

All the models work well under Dymola (2015) and OpenModelica (1.9.2).

The library will be made available open-source to the general public, if there is request.

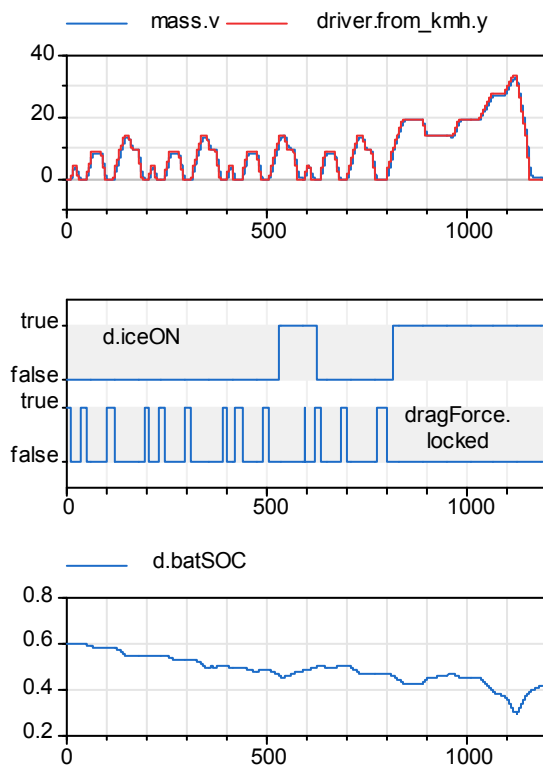


Fig. 23. Some plots related to simulation PSecu2: desired and actual vehicle speed (m/s, top); Ice-ON and vehicle Stop signals (middle), Battery SOC (dimensionless, bottom).

References

- Bondea, S. A. Nazar; "Electric Drives", CRC Press, Taylor & Francis Group, 2006 ISBN 0-8493-5220-1, section 8.17.
- UITP Project SORT: Standardised On-Road Test Cycles, 2010; data for ordering on www.uitp.org;
- M. Ehsani, Y. Gao, A. Emadi: "Modern Electric, Hybrid Electric, and Fuel Cell Vehicles: Fundamentals, Theory, and Design", CRC Press, 2009, ISBN 9781420053982
- R. Schiferl, T. Lipo: "Power Capability of Salient Pole Permanent Magnet Synchronous Motors in Variable Speed Drive Application", *IEEE Transactions on Industry Applications*, l. 26, N. 1, Jan/Feb 1990
- Toyota documentation <http://www.evworld.com> /library /toyotahs2.pdf, May 2003, retrieved from the Internet on 2015
- M. Ceraolo: "New Dynamical Models of Lead-Acid Batteries", *IEEE Transactions on Power Systems*, November 2000, Vol. 15, N. 4, pp. 1184-1190.
- S. Barsali, M. Ceraolo: "Dynamical models of lead-acid batteries: implementation issues", *IEEE Transactions on Energy Conversion*, Vol. 17, N. 1, Mar 2002, Pages 16-23.
- M. Ceraolo, T. Huria, G. Lutzemberger: "Experimentally determined models for high-power lithium batteries", Book *Advanced battery technology*, ISBN: 978-0-7680-4749-3 doi:10.4271/2011-01-1365. Also presented at the *SAE 2011 World Congress*. Cobo Center Detroit, Michigan (USA), 12-14/4/2011.

Dassault Systèmes Smart Electric Drives Library documentation: <http://www.3ds.com/fileadmin/PRODUCTS/CATIA/DYMOLA/PDF/dymola-smart-electric-drives-library.pdf>; File available for download on April 2015

Modelica Association: <https://www.modelica.org/libraries>, Vehicle Interfaces library Link available on April 2015.

EERE Information Center, https://www1.eere.energy.gov/vehiclesandfuels/pdfs/success/advisor_simulation_tool.pdf file available for download on April 2015.