Modelica'2008

Modelling of Electric Drives using `freeFOClib`

Dietmar Winkler Clemens Gühmann
Technische Universität Berlin
Chair of Electronic Measurement and Diagnostic Technology
`{Dietmar.Winkler|Clemens.Guehmann}@TU-Berlin.de`

Abstract

The `freeFOClib` (short for "free Field-Oriented Control library") provides a framework for simulations of electric drives with different application purposes. The library can be used to simply build a field-oriented control system for existing machine models from the *Modelica Standard Library*, investigate the impact of electric faults (battery faults, inverter faults, machine faults) on a electric drive system, and run simulations to estimate the fuel consumption of hybrid electric vehicles. The library structure and some of its main components are presented. Simulation results of an electric fault are given as an application example of this library. The `freeFOClib` will be publicly available in Spring 2008.

Keywords: Modelica, free library, electric machines, field-oriented control, fault simulation, hybrid electric vehicle

1 Introduction

In automotive applications the number of electric motors used is increasing rapidly. Most of them are doing their work without us – the car owners/users – actively noticing it. When a power window is still a quite obvious application for an electric motor, the active controlled throttle valve might not be. And with more and more tasks going to be performed *by-wire* (e.g., braking, steering) the number of electric motors used is due to increase even more. But not only *small* electric motors are present. With the electric motor being used for active propulsion in hybrid electric vehicles (HEV) also the power rating of motors used grows bigger.
But how do all these little and large motors work together? How should the manufacturer develop the controller? What happens if there is a fault in the system? Will the faults cause serious damage or just minor inconveniences?
All these question could be answered by using simulations to investigate the normal and faulty behaviour. For the creation of such simulation models we need both, appropriate machine models and the suitable machine controllers. The overall simulation model will contain signals from different physical domains (i.e., electrical and mechanical).
The modelling language Modelica[1] was especially developed to simplify the simulation in different physical domains in one simulation model. The multi-domain capability allows us to build simulation models of hybrid electric vehicles easier than than with other simulation tools.
So far our Chair of Electronic Measurement and Diagnostic Technology has investigated different aspects of Modelica with real-time applications and hybrid electric vehicles ([1, 2]).

2 Purpose of the library

The Modelica language is specified in the so called Modelica Specifications [3] and comes with the free *Modelica Standard Library* (MSL) [4] which contains a huge collection of models for different physical domains (e.g., electrical, mechanical, thermodynamical). For the simulation of machines the *Modelica Standard Library* contains a sub-library called `Modelica.Electrical.Machines` [5]. This library contains basic three-phase models of asynchronous and synchronous machines as well as DC machine models. To control these machines the modeller still has to provide his own controllers since currently there is no *free* Modelica library available to provide complete electric drive models.

[1]Modelica® is a free modelling language developed by the Modelica Association → www.modelica.org

So in order to simulate more complex electric drive applications a new "free Field-Oriented Control Library" (`freeFOClib`[6]) is being developed.

3 Library structure

The `freeFOClib` should allow the user to model and simulate all aspects of the an electric drive. A standard electric drive normally consist of components like power sources, power electronics, controllers, electric machines, and interfaces. The communication of the blocks can be done either via the classic approach by use of input and output connectors or by the use of bus signals. The bus structure orients itself on the new bus structure of the Vehicle Interfaces Library [7]. This should allow easier simulation of power-train simulations of hybrid electric vehicles, for example.

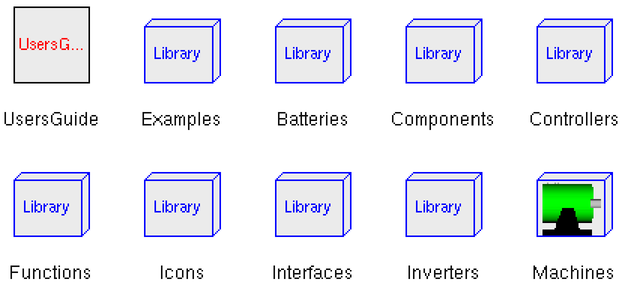

Figure 1: Top-level packages of the freeFOClib

Figure 1 shows a graphical representation of the uppermost hierarchy level of the library.
The library consists of:

`UsersGuide` Every Modelica library should contain this. It gives the user information on how to use the library as well as some information about the release history and participating developers.

`Examples` To get the user going some example simulation models are included. There are sub-packages of examples for the different parts (e.g., complete drive systems, batteries, inverters, machines)

`Batteries` This is a sub-package that contains different battery models.

`Components` Models which are library-wide used and therefore do not fit exclusively into any of the other sub-packages are placed here.

`Controllers` In here controllers for the control of electric drives together with the necessary flux models are placed.

`Functions` Custom functions which are used by the `freeFOClib`.

`Icons` Special icons for the models.

`Interfaces` The Interfaces sub-package includes different interface models. Mostly they are made of a partial type so that one can simply extend from the interface model which fits the application most.

`Inverters` These models are used to transform the control signals into electrical signals which can then be applied to the machine models.

`Machines` This library contains models of synchronous and asynchronous machines of different types.

In addition to the different controller types for the field-oriented control as well as the battery models (important for automotive applications) also new machine models for fault-simulations have been developed. These are using the m-phase presentation where the faults can be introduced directly into the components without the need of a $d-q-0$-transformation (see also [8]).

4 Library contents

After a first quick overview of the library we like to explain some the library's content in more detail.

4.1 Batteries

In the current version of the freeFOClib there are two different types of batteries present. One very simple model consisting of an internal resistor and a controlled signal voltage only. As this is normally not sufficient for more advanced simulations (e.g., simulations of driving cycles of hybrid electric vehicles) an advanced battery model was added. This model is based on models from the *Advisor 2002 Simulink®* model. It contains three main sub models: `SOC`, `VocRint`, and `BatteryECU` (see Figure 2).
The advanced battery model includes an active energy management. The state of charge is calculated by `SOC`. The simple internal resistor of the ideal resistor is replaced by a variable resistor. The value of resistance is

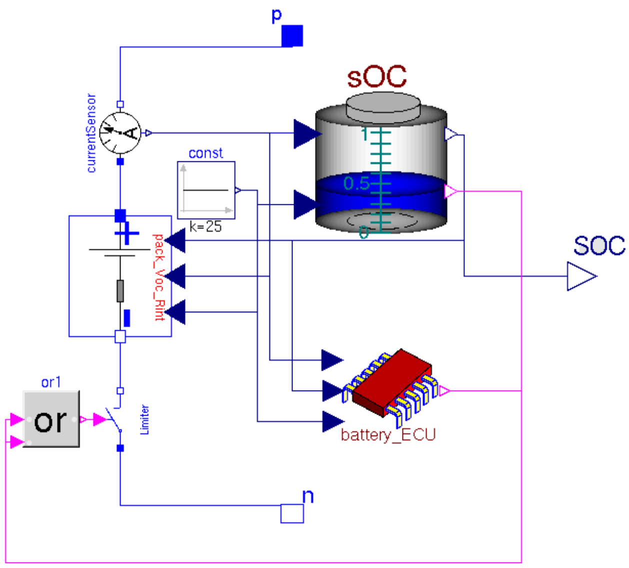

Figure 2: Advanced battery model

controlled by look-up tables and depends on the charging/discharging current, the state of charge, and the temperature. Currently the characteristic behaviour of a Nickel-Metal-Hydrid battery type is implemented. Also included is a switch which acts as a battery protection system and is controlled by the `BatteryECU` and the `SOC` models.

4.2 Controllers

This package provides models for the regulation of the control variables flux, speed, torque, and current. There is a series of different control strategies. Which of these strategies is the best applicable depends on the application. In field-oriented control the used flux-model represents a crucial part of the controller. Depending on the machine and the operational range of speed the correct flux model has to be chosen.
The `freeFOClib` offers a small variety of flux-models (see, for example, [9, 10] for more information on control of electric drives):

- Current models ($I-\vartheta$, $I-\omega$)
- Current-voltage models (UI, $UI-\omega$)
- Voltage models

4.3 Inverters

This sub-package contains different inverter models. For a start there is an `IdealInverter` which can be used if effects of the power electronic circuitry are not of interest. This gives the ideal voltage and current signals and does not need much computing power.
When effects of the power electronic are of interest than there are two different types of voltage controlled inverters available:

Space Phasor modulation Depending on the control signal an appropriate voltage space phasor is calculated. The position of the space phasor can then be transformed into firing signals for an m-phase inverter bridge (where m could be any number of phases ≥ 3).

Sinus-Delta modulation Here a sin-wave signal is compared with a high-frequency triangular wave. A logic circuitry then generates the firing signals for the m-phase inverter bridge.

Both non-ideal inverter types have the drawback that they need a lot of computational power. This has caused by the need to restart the numerical solver whenever an event (e.g., switch from one inverter leg to the other) is triggered.

4.4 Machines

Fortunately, if a modeller tries to build a standard electric drive with standard 3-phase machines he can just pick them out of the `Modelica.Electrical.Machines` library. These standard machines can be used in combination with our controllers and inverters.
Unfortunately, there is a restriction on the number of phases (i.e. $m = 3$). The `freeFOClib` provides asynchronous and synchronous induction machine models for m-phases and different architectures. The sub-package for synchronous machines contains m-phase models with electrical excited and permanent magnet rotors. The sub-package for asynchronous machines contains m-phase models with squirrel-cage and slip-ring rotor. In each of the models the stator inductance can be changed during simulation time to investigate fault impacts.
In contrast to the machine models from the *Modelica Standard Library* the m-phase models of the `freeFOClib` are not modelled in the so called $d-q-0$-frame but in the m-phase system. This might seem odd since the $d-q-0$-frame was actually introduced to reduce the computational demand of machine simulations. However when developing the `freeFOClib` one of the requirements was to be able to simulate faults and therefore unsymmetrical systems. As soon

as an electrical m-phase system becomes unsymmetrical all the computational benefit of the $d-q-0$ transformation is lost. More about the underlying theory of the machine models can be found here [8].

5 Library applications

The development of the `freeFOClib` was started with specific purposes in mind:

- field-oriented control of induction machines
- fault-simulations to investigate electrical and mechanical impacts of machine faults
- state of charge estimations for batteries in HEV applications
- investigate adaptive controller algorithms for electric machines

In automotive applications, for example, often the term drivability of a car is used. With drivability the car manufacturers often relate to the overall operating qualities of the power train. This could include things like idle mode characteristics, throttle response, and acceleration capability. In a hybrid electrical vehicle for example we got an electric motor acting directly or indirectly on the drive train.
The `freeFOClib` contains an example model which allows for simulation of three different types of faults, i.e., faults of the battery, faults in the inverter, and faults in the machine (see Figure 4).

5.1 Inverter and battery faults

To simulate faults in the battery and/or in the inverter, the example model in Figure 4 contains a fuse component `fuse_DC`. This model disconnects the DC current when a surge current is detected that violates the maximum rating of the battery. The surge-proof fuse is not triggered right away but after a first order delay time which can be parametrised. Whenever the fuse is triggered the inverter gets a signal which will switch of the firing signals for the inverter bridge in turn.
Another kind of battery fault would be a short circuit of the supply side of the inverter. This is accomplished by a simple switch that is triggered by a boolean signal and which connects both support voltage connectors of the inverter.
And at last a fault of firing signals can be applied directly via a signal `inverter_fault`. The combination of the different switches gives the abbility to build even more fault scenarios.

5.2 Machine faults

In the electric machine models of our library the following fault scenarios can be simulated:

- open-circuit of a stator phase (e.g., a connecting cable is broken)
- short-circuit phase to ground (e.g., insulation failure because of mechanical damage)
- short-circuit of one or more phase windings (e.g., insulation failure because of thermal stress within the stator or rotor)

Each of these faults will have some influence of the torque produced by the electric drive.

5.2.1 Short-circuit phase to ground

In Figure 3 you can see the simulation results of a synchronous machine with an electric excited rotor. At the time of $T = 2\,sec$ one stator phase is connected to ground. The figure shows all three phase currents and the mechanical torque over time.

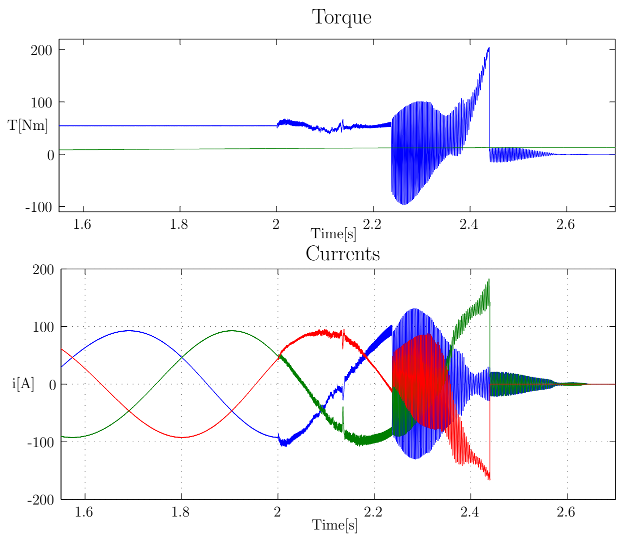

Figure 3: Fault scenario: short-circuit of one stator phase to ground

At the beginning the electric machine runs at a constant speed and with a constant load torque applied to the shaft end. When the short-circuit occurs the controller tries to keep the torque at a constant level but can only do so for a certain amount of time until the fuse finally gets triggered because of over-current. With such kind of simulation model one could for example try to find the optimum kind of fuse which

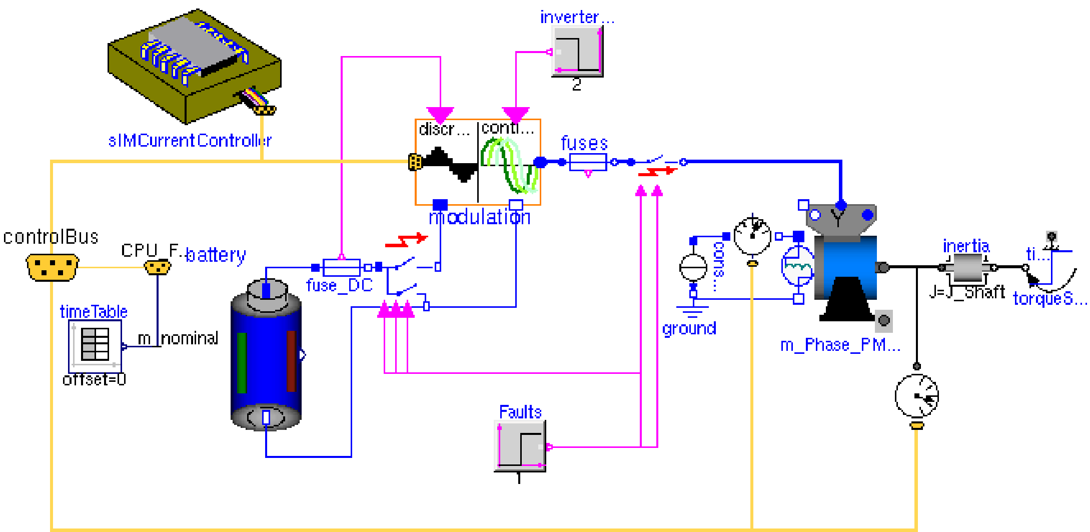

Figure 4: Example model of an electric drive system.

is sluggish enough to withstand short power surges whilst still protecting the drive's power electronic.

5.2.2 Short-circuit within a phase winding

In Figure 5 a fault of the insulation between the phase windings of a stator coil is modelled. Such a fault can be caused by, for example, over-temperature or overload which in turn leads to an overheated stator winding. This behaviour is modelled by reducing the inductance value abruptly by 20 percent. In Figure 5 you can see the three phase currents and the mechanical torque over time just before and after the connection of one phase was opened at the time of $T = 2\,sec$.
At first sight the electrical impact seems not to be very drastic. However since the field-oriented control now calculates the wrong control values the torque starts to oscillate quite considerable. If this electric drive is applied in a hybrid electric vehicle, for example, this could lead to reduction of drive comfort. But not only this, depending on the mechanical system such oscillation could become unstable and cause major damage.

5.3 Field-oriented control loop

Having a variety of different flux-models available allows the modeller to investigate different kind of control strategies for electric drives. So for example by varying some of the controllers parameters (e.g., the machine rotor resistance) one can test how robust the drive control behaves at a certain rotor speed when using different flux-models for the estimation of the flux position.

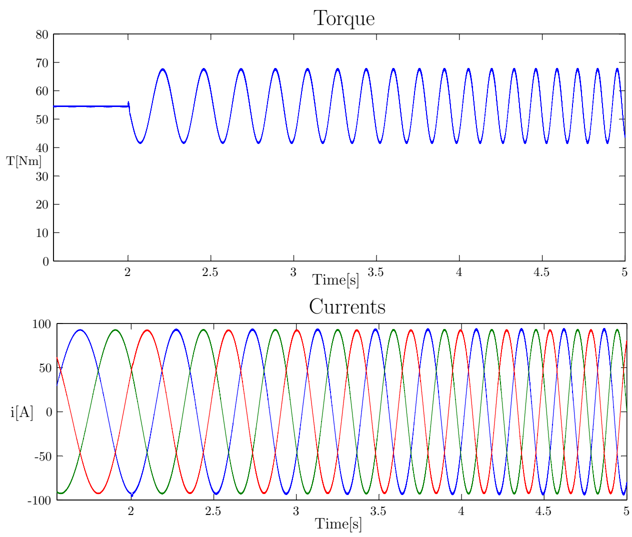

Figure 5: Fault scenario: short-circuit of phase windings

6 Future work

The main task for the future is the extension of the `freeFOClib`. Here is a short collection of ideas we are currently having:

- further clean up of the structure so that the usage is more intuitive
- add more controller types which allow also the investigation of different control architecture not just pure parameter variations
- enhance battery models with temperature models

(at the moment only look-up tables are used to calculate the resistance)

- verification of machine models
- investigate simplifications to make the machine and inverter models real-time capable

A pure estimate of the simulated values might give some clues on the general behaviour during faults. However to actually use the simulation to gain useful information (e.g., for programming a controller for the power electronics) we need more than just estimates. So the simulation model has to be verified by doing real measurements using a real electric motor. It is planned to set up test-bench system consisting of a asynchronous induction machine of from currently available hybrid electric vehicle and an electric load machine.
The first public official (pre)release is due in spring of 2008. There will be a public development repository available. For any news on the `freeFOClib` see `www.freefoclib.org`. On that site a mailing-list is also available to keep you up to date automatically.

7 Acknowledgements

We hereby like to acknowledge that a lot of work on the library was done by students as part of their student research projects:

- Stefan Rinderer, *"Development of a Modelica Library for the Field-Oriented Control of Electric Machines"*, student research project, supervisor: Dietmar Winkler, Chair of Electronic Measurement and Diagnostic Technology, TU Berlin, 16th April 2007
- Eduard Bakhach and Florian Döring, *"Modelling of a Field-Oriented Synchronous Inductionn Machine with Modelica"*, student research project, supervisor: Dietmar Winkler, Chair of Electronic Measurement and Diagnostic Technology, TU Berlin, 31st August 2007

References

[1] D. Winkler and C. Gühmann, "Synchronising a Modelica Real-Time Simulation Model with a Highly Dynamic Engine Test-Bench System," in *5th International Modelica Conference* (D. C. Kral and A. Haumer, eds.), vol. 1, (Vienna), pp. 275–281, Modelica Association, arsenal research, 4-5 Sep 2006.

[2] D. Winkler and C. Gühmann, "Hardware-in-the-Loop simulation of a hybrid electric vehicle using Modelica/Dymola," in *The 22nd International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium & Exposition* (E. . Secretariat, ed.), (Yokohama, Japan), pp. 1054–1063, EVS, Japan Automobile Research Institute, 23-28 October 2006.

[3] Modelica Association, *Modelica®– A Unified Object-Oriented Language for Physical Systems Modeling – Language Specification*, version 3.0 ed., 5th September 2007.

[4] Modelica Association, *Modelica®- Free library from the Modelica Association*, version 3.0 ed., 2008.

[5] C. Kral and A. Haumer, "Modelica libraries for dc machines, three phase and polyphase machines," in *Proceedings of the 4th International Modelica Conference* (G. Schmitz, ed.), pp. 549–558, Modelica Association, March 7-8 2005.

[6] D. Winkler, E. Bakhach, F. Döring, and S. Rinderer, "`freeFOClib` - A free Field-Oriented Control library for Modelica." unreleased, see `www.freefoclib.org` for any news. Due in, Spring 2008.

[7] M. Dempsey, M. Gäfert, P. Harman, C. Kral, M. Otter, and P. Treffinger, "Coordinated Automotive Libraries for Vehicle System Modelling," in *Proceedings of the 5th International Modelica Conference* (D. C. Kral and A. Haumer, eds.), vol. 1, (Vienna), pp. 33–41, Modelica Association, arsenal research, September 2006.

[8] D. Winkler and C. Gühmann, "Modelling of Electrical Faults Using Modelica," in *Proceedings of 'The 48th Scandinavian Conference on Simulation and Modeling (SIMS 2007)'*, Scandinavian Simulation Society, 30th - 31st October 2007.

[9] W. Leonhard, *Control of Electrical Drives*. Springer, 3rd ed., 2001.

[10] D. Novotny and T. Lipo, *Vector Control and Dynamics of AC Drives*. Clarendon Press Oxford, 1996.