

# Control and Characteristic Map Generation of Permanent Magnet Synchronous Machines and Induction Machines with Squirrel Cage

Marco Kessler  
Vorarlberg Univ. of Appl. Sc.  
Austria  
[marco.kessler@modelon.com](mailto:marco.kessler@modelon.com)

Markus Andres  
Modelon GmbH  
Germany  
[markus.andres@modelon.com](mailto:markus.andres@modelon.com)

Thomas Schmitt  
Modelon GmbH  
Germany  
[thomas.schmitt@modelon.com](mailto:thomas.schmitt@modelon.com)

## Abstract

Due to different requirements for simulation during the design process of an electric drive system, investigations were carried out to supply models of different stages with consistent sets of parameters. Therefore physical models of the Modelica Standard Library were equipped with state-of-the-art control strategies to operate in realistic conditions. In the presented case the losses computed by the physical model were stored in characteristic tables to speed up simulation in cases where dynamics are of minor interest. These models can then be used for energetic, thermal and life-time analysis with a consistent set of parameters generated from their physical counterparts.

*Keywords: PMSM, induction machine, characteristic maps, field oriented control, MTPA*

## 1 Introduction

The continuous progress in the electrification of the powertrain in the automotive industry requires numeric based computer simulations to handle the growing complexity. During development various types of simulations are needed, some of which are summarized in Table 1. Simulations of category 1 focus on brief events and need very detailed models for reliable predictions. In category 2 a tradeoff between accurate and fast models must be found, whereas category 3 and 4 especially require models which are optimized on computation time due to the long simulation runs.

The machine models of the Modelica Standard Library (MSL), which are described in detail in [3] and [1], are based on physical equations using space phasor theory. They consider leakages of the magnetic field with stray inductances and include basic models for the copper, stray load, core and friction losses.

| Category | Objectives  | Simulated time                        |
|----------|---|---------------------------------------|
| 1        | switching operations<br>nonlin. in the drivetrain               | 0.01 s – 1 s                          |
| 2        | dynamic events<br>short-term performance<br>controller behavior | 0.1 s – 100 s                         |
| 3        | energy consumption<br>thermal behavior<br>performance analysis  | 10 s – $1 \times 10^4$ s              |
| 4        | aging simulation<br>life-time of components                     | $1 \times 10^6$ s – $1 \times 10^9$ s |

Table 1: Categories for simulation

Hence, they are well suited for simulations of category 2. However, for their application in the electric powertrain control is required, which is offered in the commercial Smart Electric Drives Library (SED), but not in the MSL. For long simulation runs the SED Library also provides quasi stationary machine models [6], but these models only consider ohmic losses. Therefore, this paper describes the implementation of advanced control for the MSL machine models of the permanent magnet synchronous machine (PMSM) and the induction machine, to use them for the creation of characteristic maps for lookup table based map models. These map models include all losses which are modeled in the MSL machines in tables and allow significant improvements of the required computation times compared to the physical models. This advantage basically results from the negligence of dynamics caused by control and machine physics.

## 2 Control of the PMSM

The principle for field oriented control of the PMSM is to achieve a control concept known from the elec-

| Name                | Unit  | Description                         |
|---------------------|-------|-------------------------------------|
| $i_d$               | A     | Direct current of PMSM              |
| $i_q$               | A     | Quadrature current of PMSM          |
| $i_{sx}$            | A     | Current for field generation in IM  |
| $i_{sy}$            | A     | Current for torque generation in IM |
| $\omega_{nom}$      | rad/s | Nominal angular velocity            |
| $U_s$               | V     | Norm of stator voltage              |
| $U_{sMax}$          | V     | Norm of max. allowed stator voltage |
| $\tau_{electrical}$ | Nm    | Electrical torque inside machine    |

Table 2: Variables used in the PMSMs control algorithm

trically excited DC machine, where the magnetic flux and the torque are controlled separately. The important variables in this section are shown in Table 2.

### 2.1 The Control Scheme

As this paper focuses on the computation of losses we will not discuss the control theory in detail. Still it is important to notice that the control scheme has major influence on the generated losses. Therefore a number of effects extending the basic concept of field-oriented control have been implemented in the machine control. These are

- Voltage limitation
- Current limitation
- Field weakening<sup>1</sup>
- Maximum torque per ampere

Although these effects influence losses, it is not in the scope of this paper to review how losses are changed based on different implementation of the control algorithm. An overview of the applied scheme is presented in Figure 1.

### 2.2 Simulation Results

Figure 2 depicts the obtained simulation results where a speed step of  $2 \cdot \omega_{nom}$  was applied to a permanent magnet synchronous machine. One can see in Figure 2 b) how  $U_s$  increases proportional with the speed until  $U_{sMax}$  is reached. Then the field weakening controller starts to decrease  $i_d$  and the speed continues to rise at an expense of a decreasing the inner torque  $\tau_{electrical}$ . In Figure 2 c)  $i_q$  is plotted over  $i_d$ , where the current vector moves along the maximum

torque per ampere (MTPA) trajectory and then follows the current limit circle.

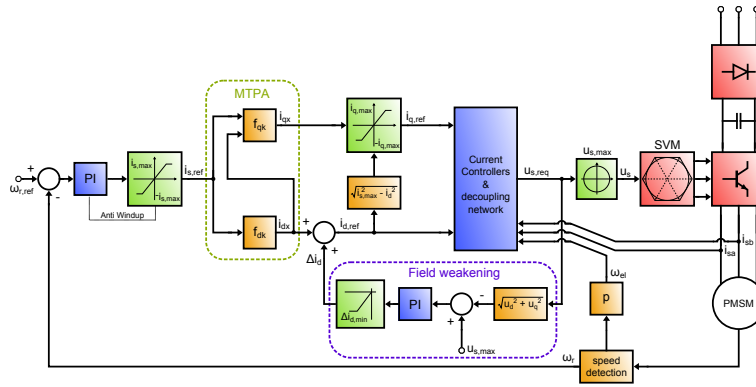
## 3 Induction machine

The control of the induction machine follows similar principles as it was presented for the PMSM. The stator voltage equation is used to create a decoupling network and to parameterize the current controllers. One difference is, that in general the rotor flux oriented reference frame is used to obtain separate control of the flux and the torque, which requires an estimator for the non-measurable rotor flux. The other difference is, that the flux must be generated, which is performed with the current component  $i_{sx}$ , whereas the  $i_{sy}$  is used for the torque generation. Differing from the PMSM the coordinate system is based on the  $x$ -axis that is aligned with the direction of the flux, whereas the  $y$ -axis is orthogonal to that and therefore responsible for generating the torque. They are named differently from the PMSMs coordinate system as the flux direction is moving at different speed from the rotors angular velocity. For more detailed informations it is referred to [5, Ch. 2 and 4.1.1] and [4].

In Figure 3 the control scheme of the induction machine is shown, which features voltage limitation, a field weakening controller and a controller for maximum torque in the upper field weakening region, which limits the output of the speed controller. For both machines, the PMSSM and the induction machine a torque controller was implemented additionally to the presented speed controller.

From the simulation results, illustrated in Figure 4, one can see how the rotor flux is generated with  $i_d$  in the first 0.5s. Then the speed increases until the voltage limit is reached, which requires to reduce the magnetizing current shown at the bottom right, which is used to create the rotor flux. As shown in the simulation results in Figures 2 and 4, a reliable control algorithm for both machine types was implemented. Thanks to the field weakening controllers and the output limitations it is ensured that the voltage and current limitations are not exceeded. Hence, the controlled machine models can be used in batch simulations in order to compute the characteristic map for a given set of machine parameters as well as stator current and stator voltage limitations.

<sup>1</sup>The algorithm used in this paper is based on a comparison of  $U_s$  and  $U_{sMax}$  and is based on theory presented in [4]



Source: Based on [2]

Figure 1: Extended control scheme for the PMSM with MTPA and field weakening

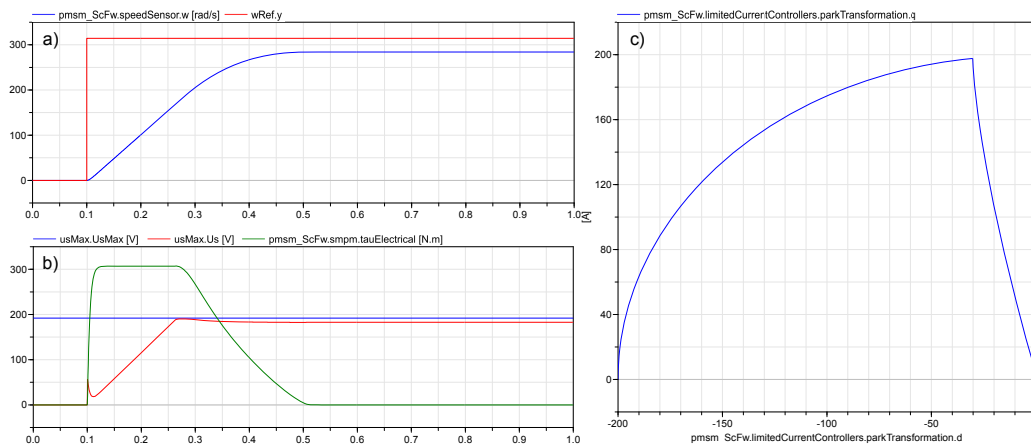


Figure 2: Simulation results when a speed step is applied to a speed controlled PMSM with field weakening (*pmsm\_ScFw*). (a) rotor speed reference and actual value; (b) voltage limit, measured voltage and electric torque; (c)  $i_q$  over  $i_d$

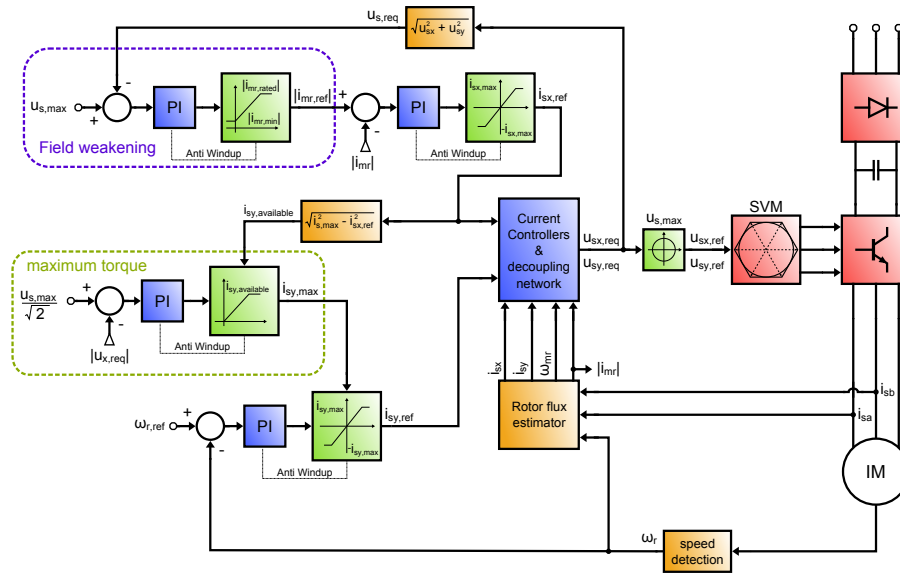
## 4 Characteristic Maps

If a driving cycle is simulated for a hybrid or an electric car the energy consumption of different parts and the resulting attainable range are usually most important, whereas quantities like the magnetic flux or the voltage drops at the stray inductances are of minor interest. Therefore it is not required to have detailed physical models and the use of averaged models is sufficient. For electric machines such models can be composed of characteristic maps, in which the losses at various operating points are stored.

Characteristic maps of machines can also be used if the losses of a real machine are measured in different operating points. Then a convenient way to have an accurate model of this machine is to use the measurement results in the simulation within lookup tables. However, in an early design stage the real machine might not be available. Therefore in this paper

a method is presented which allows to calculate characteristic maps from simulations which are performed with physical models of a machine. In the paper the machine models from the Modelica Standard Library are used, but they can be replaced with more or less accurate models. This possibility is a major advantage compared to other methods e.g. averaged models that do simplifications to the models to make it compute faster. This way the modeling detail can be increased to a very high level only influencing the map generation time. As this has to be carried out once only for every new machine it is of minor importance compared to the many times the map-based model is simulated.

In the following sections it is shown how the maps can be generated and lookup table based models are presented which utilize those.



Source: Based on [4]

Figure 3: Control scheme for the induction machine

## 4.1 Generation of Maps

To generate characteristic maps, models are used which utilize the controlled machine models presented in the preceding sections. In addition, a Modelica function was created, which simulates the machines in different operating point and stores the resulting losses in a .mat file, which can later on be used in lookup tables.

### 4.1.1 Model for Map Generation

In Figure 5 the model for the characteristic map generation of the induction machine is shown. For PMSMs a similar model was created. The green box “controlled machine” contains a torque controlled induction machine and a data record for machine parameters. With the block torque the desired torque is set. In the box “fixed speed” the block speed is used to set the angular velocity at the shaft of the machine model. The machine is accelerated to the desired speed and due to the torque controller in the machine model, the requested torque is obtained at the shaft as long as the current limit is not reached.

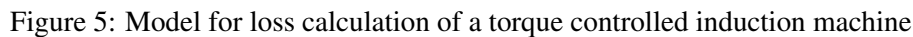
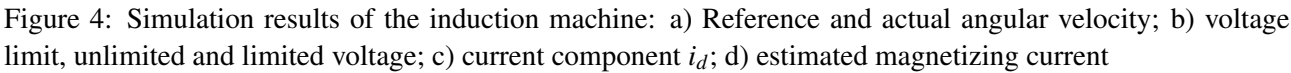
In the box “end Simulation” it is checked if the operating point is reached and if the losses are settled. When this is the case, the simulation is terminated successfully with the terminateSimulation block. For this purpose the block *WithinLimit* was created, which outputs true only if the input is within the specified limit. This block is used for torqueReached, torqueSettled and lossesSettled. It is verified

that the measured torque at the flange matches the desired torque and that the total losses of the MSL are settled.

In Figure 6 simulation results of this model are depicted. The simulation finishes successfully after 1.3 s, since all required conditions are fulfilled. One can observe a slight drop of the losses after the final speed is reached. Since the acceleration is finished, the stator current and in turn the torque is reduced. Hence, one must take care that the simulation is not aborted too early, since this would result in too high losses for this operating point. With the block lossesSettled this is prevented.

### 4.1.2 Function for Map Generation

For the map generation a function is provided, which utilizes the model *TorqueControlledPhys2Map\_ASM* shown in Figure 5 to calculate a characteristic map with a specific parameter set. The user has to specify a curve, which defines the maximum torque the machine is capable to deliver for the speed range which should be covered with the map. This curve is used as the upper limit for the performed simulations. To include generator and motor mode, the torque curve can be defined for both negative and positive angular velocities. For the generated maps it is intended that the maximum torque and the losses are independent of the direction of rotation. Thus, the same values are used for both rotational directions of motor and generator mode. In addition to the torque curve, the user



must specify the resolution of the map, which is used to determine which operating points will be simulated.

The calculated losses  $P$  of the simulated operating points are stored in matrices, with the first column containing the torque and the first row the angular velocity, resulting in

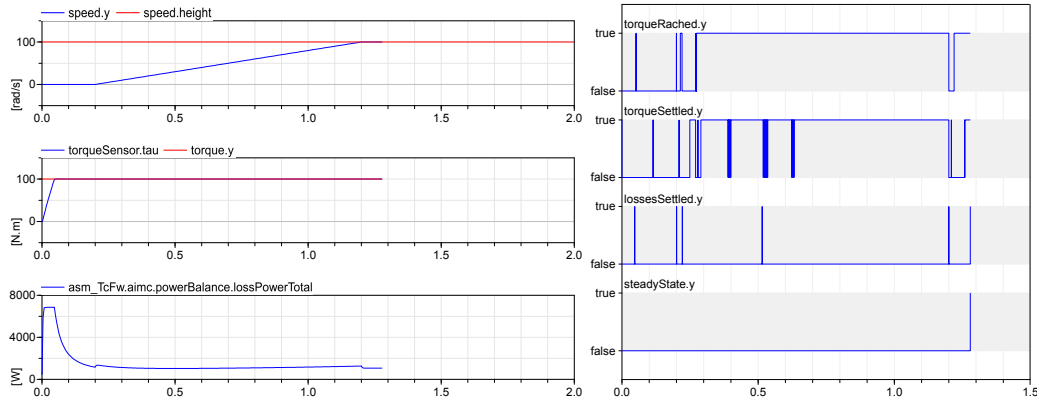
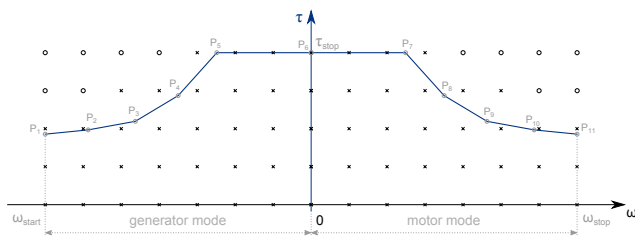
$$\begin{bmatrix} 0 & \omega_{start} & \omega_2 & \dots & \omega_{stop} \\ 0 & P_{1,1} & P_{1,2} & \dots & P_{1,y} \\ \tau_2 & P_{2,1} & P_{2,2} & \dots & P_{2,y} \\ \tau_3 & P_{3,1} & P_{3,2} & \dots & P_{3,y} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tau_{stop} & P_{x,1} & P_{x,2} & \dots & P_{x,y} \end{bmatrix}. \quad (1)$$

The ascending values in the first row and column are required for the usage in the lookup tables *Modelica.Blocks.Tables.CombiTable2D*. The range for the angular velocity is given by the lowest and the highest entry

of the user. The torque is simulated from zero up to the highest torque value which was entered. The area in between is discretized according to the user chosen resolution.

In Figure 7 an exemplary maximum torque curve and the performed discretization are illustrated. Due to arbitrary user inputs for the torque curve and the discretization, it is not possible to always simulate exactly up to the maximum torque curve. Hence, to cover the desired region it is required to perform one simulation above the curve. This is also illustrated in Figure 7, where simulated points are marked with  $\times$ , whereas for non-simulated points circles are used.

Since the whole matrix presented in Equation 1 has to be filled with values, the points marked with circles in Figure 7 have to be guessed somehow. To do so, the function *Modelica.Math.Vectors.interpolate* is used to


Figure 6: Simulation results of *TorqueControlledPhys2Map\_ASM*

Figure 7: Exemplary maximum torque line and the performed discretization for a map of  $5 \times 15$ 

perform linear extrapolation from simulated values, which it does although the name would not indicate that. The extrapolated values for not simulated operating points in Equation 1 are calculated from the two rows above. This and the linear extrapolation might not be suited very well to calculate the correct values, but since it is used for the area above the maximum torque curve, the machine will hardly enter this region and it is still a much better approximation compared to leaving the losses on zero.

## 4.2 Model to utilize Characteristic Maps

To use the calculated maps in simulations, the model *TorqueControlledCharacteristicMap* is used, which is illustrated in Figure 8. It features two electrical DC connectors to the left, a mechanical connector to the right and a real input for the desired torque at the top. Hence, it is meant to replace a controlled electric machine with inverter. Via a parameter the .mat file containing the losses is specified.

The torque reference given with the real input *desiredTorque* is limited by *limitTorque* according to the maximum torque line, which the user has defined during map generation. Thereafter the first or-

der element *firstOrder* introduces a delay between desired and obtained torque<sup>2</sup>. The first order element is connected to the torque source *mechanicalTorque*, which accelerates *rotorInertia*. Parameters are provided in the model, which allow the user to deactivate *limitTorque* and *firstOrder*. Then the gain blocks to *noTorqueLimit* and *noFirstOrder* are used instead.

In the violet colored box “Losses” one lookup table of the type *CombiTable2D* from the MSL is used for each loss type. By knowing the actual speed and torque, the operating point is identified and the losses of this point can be determined with the tables. Afterwards the electrical power is calculated from the sum of the losses and the measured mechanical power. This allows the calculation of the electrical current that must flow by measuring the voltage at the DC connectors.

## 4.3 Comparison Results

The hardware, software and solver settings which were used for the measurement of the computation time and the required time for map generation are summarized in Table 3.

### 4.3.1 Computation Time for Map Generation

The required computation time for the generation of characteristic maps was tested for each machine type for the two different map resolutions  $15 \times 30$  and  $30 \times 60$ . In Table 4 the results are shown. The column “unsim.” notes the number of skipped simula-

<sup>2</sup>The element is intended for the torque set machines to model the electric time constant. For torque controlled machines it is disabled per default, but it can be used to approximate the transfer function of desired to obtained torque.

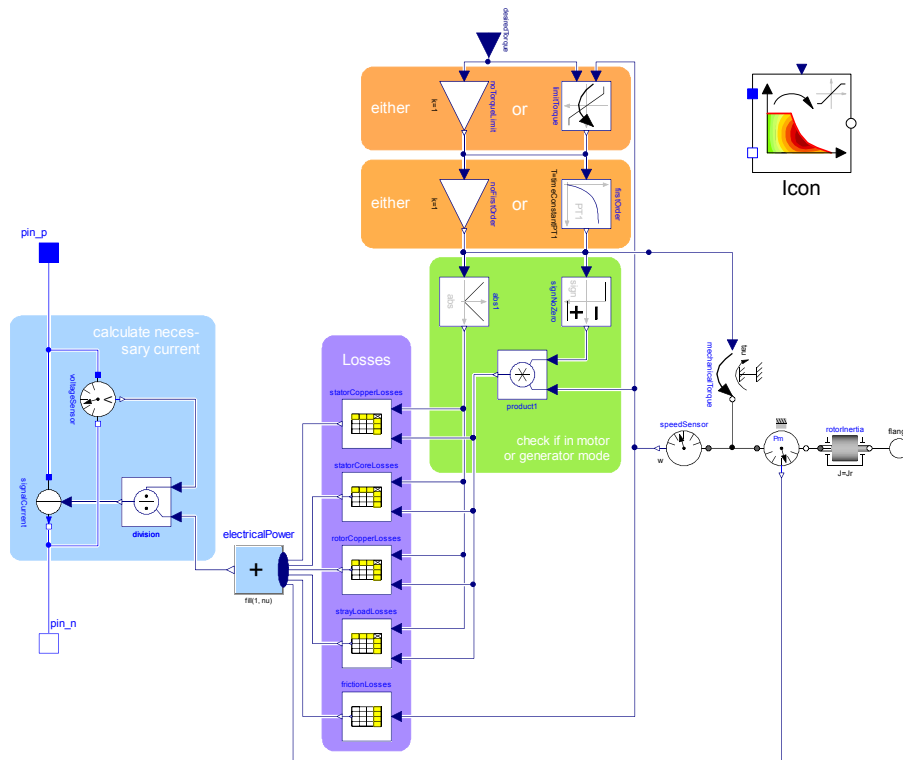


Figure 8: Model and icon for the torque controlled characteristic map

|          |           |                      |
|----------|-----------|----------------------|
| Hardware | Processor | Intel Core i7-2670QM |
|          | RAM       | 8 GiB DDR3           |
| Software | OS        | Windows 7, 64 bit    |
|          | Dymola    | 2013 FD01, 32 bit    |
| Settings | Solver    | Dassl                |
|          | Tolerance | 0.0001 (default)     |

Table 3: Used hardware, software and solver settings for time measurements

| Type | $\tau$ | $w$ | unsim. | req. time | time/sim. |
|------|--------|-----|--------|-----------|-----------|
| ASM  | 15     | 30  | 204    | 190 s     | 0.78 s    |
| ASM  | 60     | 30  | 854    | 740 s     | 0.78 s    |
| PMSM | 15     | 30  | 120    | 95 s      | 0.29 s    |
| PMSM | 60     | 30  | 488    | 370 s     | 0.28 s    |

Table 4: Required computation time (req. time), skipped simulations (unsim.) and required time for one simulation (time per sim.) for the generation of characteristic maps of different resolutions ( $\tau \times w$ )

tions, where the operating points lay above the maximum torque curve.

In the column “time/sim.” it is revealed that the required time for one simulation is constant for the different machine types. Hence, maps with 1000 calculated points are generated within 13 minutes for induction machines and less than 6 minutes for PMSMs. However, the map generation time is highly sensitive to the specified limits in which the losses and the torque have to settle. As it can be noticed in Figure 6 for the plot “torque settled” to low margins can result in chattering depending on the operating point. Hence, the high number of events slows down the map generation process significantly. So far no effort was spent to further investigate on that effect, but it would be well worth the effort if map generation time is of impor-

tance.

From Table 4 one can tell that the solving of the ASM’s models is about eight times as computationally intensive than the PMSM’s. This is caused by either the higher complexity of the ASM’s model due to effects like slip that is not present in the PMSM or due to the more complex controller structure. The exact cause has to be further investigated in future efforts.

### 4.3.2 Accuracy of the Losses

To compare the losses of the map model with those of the physical model, the setup depicted in Figure 9 is used. With `desiredSpeed` and `desiredTorque` the operating point is set for the map based and the



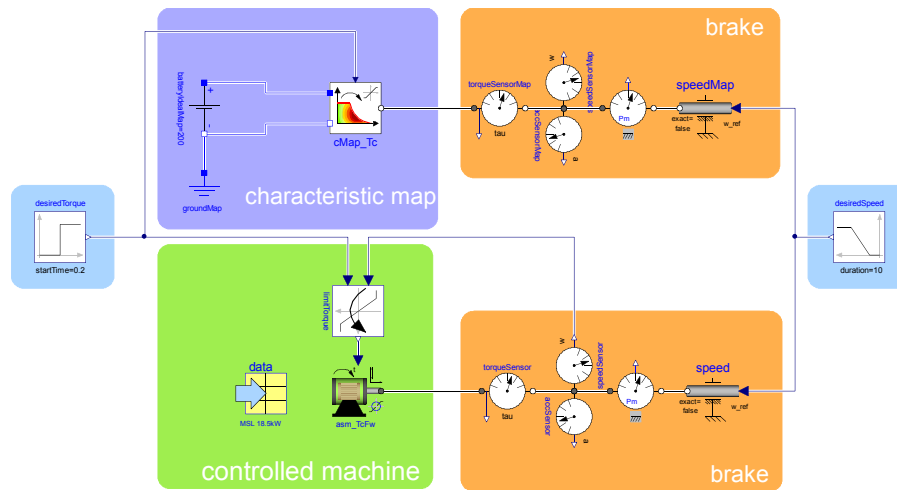


Figure 9: Used model for comparison of torque controlled map and physical machine

physical models and the input of the physical model is limited to the maximum torque curve. Similar models were used to compare the losses of the other machine types.

In Figure 10 the losses of an efficiency map are compared with the physical model. The desired torque is set on 100 N m, whereas the speed is increased from 0 rad s<sup>-1</sup> to 400 rad s<sup>-1</sup> within 10 s.

The comparison results of the torque controlled PMSM are shown in Figure 11. As one can see, the losses match very well. The plot “Deviation of total losses” at the bottom right shows the difference of the total losses calculated with the physical model and with the map model. Except of the negative peak at the beginning, the total deviation remains below 18 W. With the total losses starting at 1000 W and reaching about 1800 W when the highest deviation is observed the maximum error is 1%. The peak is caused by the stator core losses, since the current controllers set the maximum possible voltages at the beginning to obtain the desired torque. As the core losses are plotted over speed, this peak is hidden behind the axis to the left at 0 rad s<sup>-1</sup>.

#### 4.3.3 Simulation Time

To compare the computation time of physical and map-based models, the model shown in Figure 12 is used. A sinusoidal or a trapezoid torque is requested from the machine and a speed dependent torque is used as load. With the integrator the total loss energy is obtained, which allows to check for the global error of the most interesting quantity at the end of the simulation. This error is computed between physical and ta-

ble based model, not the analytic solution as the word "error" may indicate.

The results for the sinusoidal input are depicted in Table 5. For every machine type the map model and the physical machine were tested for a simulation time of 1000 s and for 10000 s. Along with the required simulation times the speed improvement factor is given. In addition, the loss energy at the end of the simulation and the error of the map model are specified.

One can see that improvements of the simulation time of more than a factor of 200 can be obtained, while the error is kept below 1 %. The highest improvements are achieved for the induction machine, while the torque set PMSM is only accelerated by a factor of 7.

## 5 Conclusion

The presented method allows convenient use of controlled physical electric machine models including advanced functionalities, to generate data for map-based models in order to accelerate simulations with still accurate results.

With the map models speed improvements with factors starting from 7, reaching up to 230. However the computation times of the physical models are highly sensitive to the performed simulation, the input signal and to the tuning of the controllers. Hence, it is required to test the models within specific applications to figure out which factors are obtained in practice.



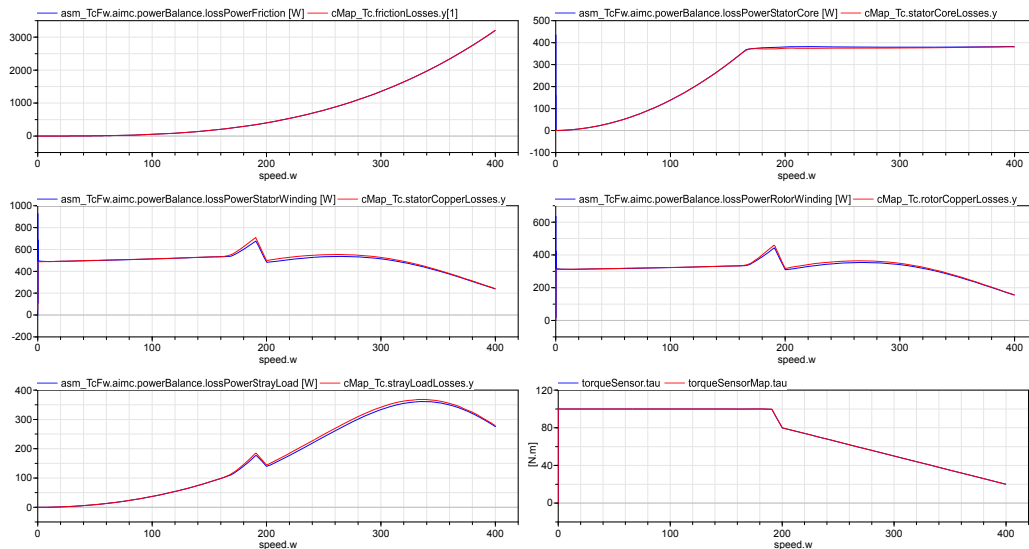


Figure 10: Losses of a physical induction machine model (blue) and characteristic map (red) with a resolution of  $15 \times 15$  for the motor mode

| Type          | CPU times |        |        | loss energy at simulation end |             |        |
|---------------|-----------|--------|--------|-------------------------------|-------------|--------|
|               | phys.     | map    | factor | phys.                         | map         | error  |
| ASM, 1000 s   | 82 s      | 0.35 s | 234    | 1753.60 kJ                    | 1763.04 kJ  | 0.5 %  |
| ASM, 10000 s  | 807 s     | 3.5 s  | 230.5  | 17537.90 kJ                   | 17633.20 kJ | 0.5 %  |
| PMSM, 1000 s  | 9.5 s     | 1.3 s  | 7.3    | 4778.35 kJ                    | 4784.76 kJ  | 0.13 % |
| PMSM, 10000 s | 95.7 s    | 13 s   | 7.4    | 47782.80 kJ                   | 47848.70 kJ | 0.13 % |

Table 5: Required computation times for map based and physical models with sinusoidal input

## References

- [1] A. Haumer, C. Kral, H. Kapeller, T. Bäuml, and J. V. Gragger. The AdvancedMachines Library: Loss Models for Electric Machines. In *Proceedings of the 7th International Modelica Conference*, pages 847–854, September 2009.
- [2] J.-M. Kim and S.-K. Sul. Speed control of interior permanent magnet synchronous motor drive for the flux weakening operation. *Industry Applications, IEEE Transactions on*, 33(1):43–48, 1997.
- [3] C. Kral and A. Haumer. Modelica libraries for dc machines, three phase and polyphase machines. In *Proceedings of the 4th International Modelica Conference*, pages 549–558, March 2005.
- [4] M. Mengoni, L. Zarri, A. Tani, G. Serra, and D. Casadei. A Comparison of Four Robust Control Schemes for Field-Weakening Operation of Induction Motors. *Power Electronics, IEEE Transactions on*, 27(1):307–320, 2012.
- [5] P. Vas. *Sensorless Vector and Direct Torque Control (Monographs in Electrical and Electronic Engineering)*. Oxford University Press, USA, 1998.
- [6] J. Vinzenz, G. Harald, and G. C. Kral. The Smart-ElectricDrives Library - Powerful Models for Fast Simulations of Electric Drives. In *Proceedings of the 5th International Modelica Conference*, pages 571–577, 2006.

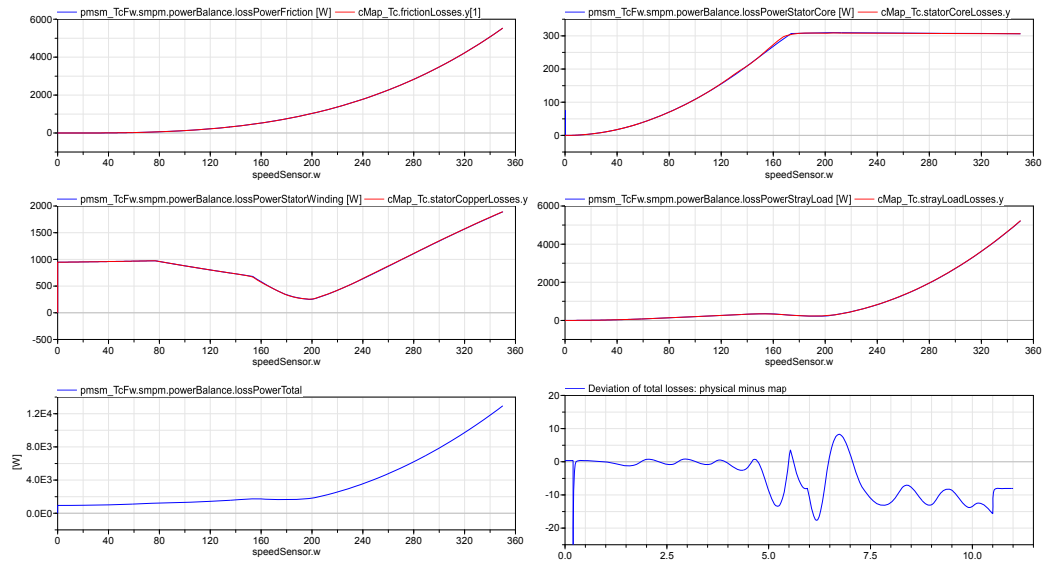


Figure 11: Losses of a torque controlled physical PMSM model (blue) and characteristic map (red) with a resolution of  $20 \times 15$  for the motor mode

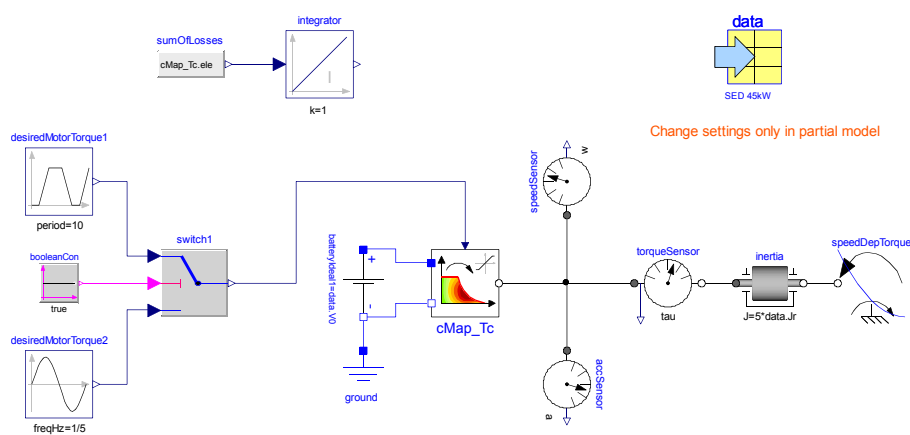


Figure 12: Model for comparison of computation times