Wind turbine blockset in Matlab/simulink. General overview and description of the models

Iov, F.; Hansen, Anca Daniela; Sørensen, Poul Ejnar; Blaabjerg, F.

Publication date: 2004

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Wind Turbine Blockset in Matlab/Simulink

General Overview and Description of the Models

Florin Iov, Anca Daniela Hansen, Poul Sørensen, Frede Blaabjerg

Aalborg University
March 2004
Abstract. This report presents a new developed Matlab/Simulink Toolbox for wind turbine applications. This toolbox has been developed during the research project “Simulation Platform to model, optimize and design wind turbines” and it has been used as a general developer tool for other three simulation tools: Saber, DlgSILENT, HAWC. The report provides first a quick overview over Matlab issues and then explains the structure of the developed toolbox. The attention in the report is mainly drawn to the description of the most important mathematical models, which have been developed in the Toolbox. Then, some simulation results using the developed models are shown. Finally, some general conclusions regarding this new developed Toolbox as well as some directions for future work are made.
# Table of contents

Table of contents .......................................................................................................................... 3  
Preface ........................................................................................................................................ 5  
Chapter 1  Introduction ............................................................................................................. 6  
Chapter 2  Matlab/Simulink® – An overview ........................................................................... 7  
Chapter 3  Wind Turbine Blockset........................................................................................... 9  
  3.1  General structure .................................................................................................. 9  
  3.1.1  Mechanical Components Library ..................................................................... 10  
  3.1.2  Electrical Machinery Library ............................................................................ 11  
  3.1.3  Power Converters Library ................................................................................ 12  
  3.1.4  Common Bocks Library ................................................................................... 13  
  3.1.5  Transformations Library ................................................................................... 14  
  3.1.6  Measurements Library ...................................................................................... 14  
  3.1.7  Control Library ................................................................................................. 15  
  3.2  Simulation speed in Matlab/Simulink ....................................................................... 16  
  3.3  Summary ..................................................................................................................... 7  
Chapter 4  Description of the models..................................................................................... 21  
  4.1  Mechanical component models................................................................................ 21  
  4.1.1  Wind model ...................................................................................................... 21  
  4.1.2  Wind turbine rotor model ................................................................................. 23  
  4.1.3  Drive trains models ........................................................................................... 25  
  4.1.3.1  Three-mass model ......................................................................................... 25  
  4.1.3.2  Two-mass model .......................................................................................... 26  
  4.1.3.3  One-mass model ........................................................................................... 27  
  4.2  Electrical machinery models.................................................................................... 28  
  4.2.1  Induction machine models................................................................................ 28  
  4.2.1.1  ABC/abc model of induction machine ......................................................... 28  
  4.2.1.2  Model of induction machine in dqo arbitrary reference frame..................... 33  
  Fluxes as state variables ............................................................................................. 34  
  Currents as state variables .......................................................................................... 35  
  Electromagnetic torque ............................................................................................... 36  
  4.2.1.3  Reduced order model .................................................................................... 37  
  4.2.1.4  Steady-state model ......................................................................................... 38  
  4.2.1.5  Modelling deep-bar effect ............................................................................ 41  
  4.2.1.6  Modelling saturation ...................................................................................... 45  
  4.2.1.7  Modelling iron losses ................................................................................... 46  
  4.2.2  Synchronous Machine Model........................................................................... 48  
  4.2.2.1  Dynamic Equations of Synchronous Machine ............................................. 48  
  Untransformed model ................................................................................................. 48  
  Transformed model with quadrature-phase stator windings .................................... 49  
  Commutator model ..................................................................................................... 50  
  Electromagnetic torque ............................................................................................... 52  
  4.2.2.2  Simulink model ............................................................................................. 52  
  4.2.3  Permanent-magnet synchronous machine ....................................................... 53
4.3  Power converter models ................................................................. 56
  4.3.1  AC Controllers (Soft-starters) .................................................. 56
    4.3.1.1  Star connected 3-phase load ........................................... 58
    4.3.1.2  Delta connected 3-phase load ....................................... 60
    4.3.1.3  Branch-delta connected load ......................................... 61
    4.3.1.4  Simulink implementation of the soft-starter ................... 62
    4.3.1.5  RMS model for soft-starter ........................................... 62
  4.3.2  Voltage Source Converters ....................................................... 64
    4.3.2.1  Star connected generator ............................................. 65
    4.3.2.2  Delta connected generator ........................................... 66
    4.3.2.3  Simulink implementation of the voltage source converter ... 67
  4.3.3  DC-link Circuit ........................................................................ 68
4.4  Modulation strategies ................................................................. 68
  4.4.1  PWM sinusoidal modulator using sine-triangle with third harmonic ... 69
  4.4.2  Space Vector Modulation ....................................................... 70
4.5  Grid component models ............................................................... 72
  4.5.1  Grid model ............................................................................. 72
  4.5.2  Circuit breaker model ............................................................ 73
  4.5.3  Capacitor bank ....................................................................... 75
  4.5.4  Cable model ............................................................................ 76
  4.5.5  Transformer model ................................................................... 78
    4.5.5.1  3-phase 2-winding transformer ...................................... 78
    4.5.5.2  Determination of ideal transformer matrix ...................... 79
    4.5.5.3  Parameter estimation for linear transformer model .......... 80
      Determination of parameters for lossless three-phase transformer ... 80
      Inclusion of loss terms ......................................................... 81
    4.5.5.4  Dynamic equations of 3 phase 2 windings transformer ..... 82
  4.5.6  Phase Locked Loop ............................................................... 83
4.6  Control blocks – Control of PQ for DFIG ...................................... 84
    Stator-flux oriented control ....................................................... 84
4.7  Summary ..................................................................................... 87
Chapter 5  Simulation results ............................................................. 88
  5.1  Connection of a no-load transformer to the grid ....................... 88
  5.2  Disconnection of a no-load transformer .................................... 89
  5.3  Start-up sequence of a fixed-speed wind turbine ....................... 90
  5.4  Variable speed/pitch wind turbine with DFIG ......................... 92
  5.5  Summary .................................................................................. 21
Conclusions and Future work .......................................................... 94
References ....................................................................................... 97
Appendix A  An example of a “C” S-Function .................................... 101
Appendix B  Other developed models ................................................. 105
    Harmonic voltage source and grid angle detection .................... 105
    Three-phase diode bridge rectifier ......................................... 107
    Synchronous PWM modulator ............................................. 107
    ABC to Magnitude and phase transformation ......................... 108
    Average wind calculator ....................................................... 108
Preface

This report describes the *Wind Turbine Blockset* developed in Matlab/Simulink during the project “A Simulation Platform to Model, Optimize and Design Wind Turbines”. The project has been funded by the Danish Energy Agency contract #ENS-1363/01-0013, and it was carried out in cooperation between Aalborg University and RISØ National Laboratory.
Chapter 1

Introduction

In the last years Matlab/Simulink has become the most used software for modelling and simulation of dynamic systems. Wind turbine systems are an example of such dynamic systems, containing subsystems with different ranges of the time constants: wind, turbine, generator, power electronics, transformer and grid.

Matlab/Simulink provides a powerful graphical interface for building and verifying new mathematical models as well as new control strategies for the wind turbines systems. Then, using a dSPACE prototyper these new control strategies can be easily implemented and tested in a Hardware-In-the-Loop structure.

This report presents a new developed Matlab/Simulink Toolbox for wind turbine applications. This toolbox has been developed during the research project “Simulation Platform to model, optimize and design wind turbines”. Matlab/Simulink has been used in this Simulation Platform as a general developer tool for other three simulation tools, namely: Saber, DlgSILENT and HAWC.

The report provides first a quick overview over Matlab issues and then explains the structure of the new developed toolbox.

A special attention has been on increasing the simulation speed for all the models developed in this new Toolbox. Some aspects regarding the implementation methods in Simulink for a dynamic model are presented.

The attention in the report is mainly drawn to the description of the most important mathematical models, which have been developed in the Toolbox.

Some simulation results using the developed models are shown.

During the work carried out in developing of this Toolbox a large amount of references including books, papers and other sources have been used. In the report only the most relevant references are mentioned for each particular topic. However, the reference list presented in report includes almost all references in alphabetical order used during the work.
Chapter 2

Matlab/Simulink® – An overview

MATLAB® is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in a familiar mathematical notation [78]. Typical uses include:

- Math and computation;
- Algorithm development;
- Data acquisition;
- Modelling, simulation, and prototyping;
- Data analysis, exploration, and visualization;
- Scientific and engineering graphics;
- Application development, including graphical user interface building.

The MATLAB system consists of five main parts [78]:

- Development Environment. This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- MATLAB Mathematical Function Library. This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

- MATLAB Language. This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

- Graphics. MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- MATLAB Application Program Interface (API). This is a library that allows to write C and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.
In the last few years, Simulink® has become the most widely used software package in academia and industry for modelling and simulating dynamic systems [78].

Simulink® is a graphical software package for modelling, simulating, and analyzing dynamic systems and it is based on Matlab®. It supports linear and nonlinear systems, modelled in continuous time, sampled time, or a hybrid of the two. Systems can also be multirate, i.e., have different parts that are sampled or updated at different rates. For modelling, Simulink provides a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. With this interface, the desired dynamic systems can be easily built. Simulink includes a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. Using S-Functions it is also possible to customize and create user-defined blocks. Models are hierarchical, so the models can be built using both top-down and bottom-up approaches. The system can be viewed at a high level, then double-click blocks to go down through the levels to see increasing levels of model detail. This approach provides insight into how a model is organized and how its parts interact. After defining a model, it can be simulated, using a choice of integration methods, either from the Simulink menus or by entering commands in the MATLAB Command Window. The menus are particularly convenient for interactive work, while the command-line approach is very useful for running a batch of simulations (e.g. Monte Carlo simulations or sweeping a parameter across a range of values). Using scopes and other display blocks, the simulation results can be analyzed while the simulation is running. In addition, the parameters can be changed during the simulation for "what if" exploration. The simulation results can be put in the MATLAB workspace for postprocessing and visualization. Model analysis tools include linearization and trimming tools, which can be accessed from the MATLAB command line, plus the many tools in MATLAB and its application toolboxes. Since MATLAB and Simulink are integrated, the models can be simulated, analyzed, and revisited in either environment at any point.

An S-function is a computer language description of a Simulink block. S-functions can be written in MATLAB, C, C++, Ada, or Fortran. C, C++, Ada, and Fortran S-functions are compiled as MEX-files using the \texttt{mex} utility [78]. As with other MEX-files, they are dynamically linked into MATLAB when needed. S-functions use a special calling syntax that enables the user to interact with Simulink's equation solvers. This interaction is very similar to the interaction that takes place between the solvers and built-in Simulink blocks. The form of an S-function is general and can accommodate continuous, discrete, and hybrid systems. S-functions allow the user to add new models to the Simulink built-in models. These new blocks can be written in MATLAB®, C, C++, Fortran, or Ada. An algorithm or model can be implemented in an S-function and then customize a user interface by using masking.
Chapter 3

Wind Turbine Blockset

In this chapter the structure of a new Matlab/Simulink Toolbox for wind turbine applications is presented. The content of the main libraries from this toolbox is briefly shown. Then, some considerations about the simulation speed for the developed models are done.

3.1 General structure

A new Matlab/Simulink Toolbox for wind turbine applications has been developed during the project. In order to analyze the dynamic and/or steady state behaviour of a wind turbine, the basic components of a wind turbine have been modelled and structured in seven libraries: Mechanical Components, Electrical Machinery, Power Converters, Common Models, Transformations, Measurements and Control as shown in Fig. 3.1.

These models are built based on Simulink blocks as well as using C S-Functions. Some of the features of this developed Toolbox can be summarized as follows:

- All the developed models use basically only Simulink blocks. No particular Toolbox is necessary to simulate a wind turbine concept. So, the user should have installed on the PC only Matlab and Simulink. However, some models uses particular blocks from other Toolboxes from Matlab e.g. Power Spectra Density calculation block;
- It uses the matrix support in order to minimize the number of blocks and connection lines;
- All models which involves a great number of differential equations (e.g. electrical machines, drive-trains and transformer) are also available as ‘C’ S-Functions for high-speed simulations;
- In order to be able to use different drive-train models the equation of motion is not included in the electrical machine models;
- Special models have been developed for induction generators (abc/abc models), which can be used in some particular fault analysis e.g. unsymmetrical, unbalanced faults with unbalanced loads. In this case cannot be used the standard dqo model. Moreover, this model is used to simulate the soft-starter fed induction machines, which are used in fixed-speed wind turbines.

A short description of the libraries illustrated in Fig. 3.1 is presented in the following paragraphs.

3.1.1 Mechanical Components Library

The first library Mechanical Components contains: wind model, aerodynamic models of the wind turbine rotor, different types of drive train models as shown in Fig. 3.2.

![Fig. 3.2. Mechanical Components Library.](image)

The wind model has been developed at RISO National Laboratory based on the Kaimal spectra. The wind speed is calculated as an averaging of the fixed-point wind speed over the whole rotor, and it takes the tower shadow and the rotational turbulences into account [71].

Since one of the main components in this model is the normally distributed white noise generator some investigations have been done in order to obtain the same wind time series in all considered simulation tools. It has been found that the built-in white noise generator from different simulation tools uses different algorithms and therefore different wind time series are obtained. A new normally distributed white noise generator has been implemented using a ‘C’ S-Function based on the Ziggurat Algorithm developed by G. Marsaglia [45].
The aerodynamic models of the wind turbine rotor are based on the torque coefficient $C_Q$ or power coefficient $C_P$ look-up table.

As mentioned before, the equation of motion is not included in the machine model. Different types of drive train models are developed e.g. one-mass model, two-mass model of the drive train with torsional torques. Since the last model involves 4 differential equations an alternative implementation using C S-Function is also available.

### 3.1.2 Electrical Machinery Library

The *Electrical Machinery Library* contains models with different level of detailing for electrical machines used currently in wind turbine systems as shown in Fig. 3.3. The standard $dq$ models as well as the new models for squirrel-cage induction machine, wound rotor induction machine, salient-poles synchronous machine and permanent magnet synchronous machines have been included in this library.

![Fig. 3.3. Electric Machinery library based on Simulink blocks.](image)

Currently, all commercial simulation software uses the $dq$- or $dqo$-reference frames in modelling the electrical machines [83]-[85]. However, some particular operating modes e.g. soft-starter-fed induction machines requires an $abc/abc$ model [67], [69], [73], [74]. Therefore, the models of the electrical machines are written in state-variable form in the $dqo$-reference frame as well as in the $abc/abc$ natural reference frame. Some special features, as deep-bar effect in the rotor of the induction machines, are also included in models. Further extensions of the models e.g. temperature, saturation, etc. can easily be added.
Another important feature of the abc/abc machine models is that the dynamic equations are written in per-phase quantities. Therefore, the desired winding-connection can be obtained e.g. star, delta or branch-delta as in the case of soft-starters-fed induction machines.

Besides the complete dynamic models of induction machine some reduced order models have been developed and implemented in the advanced aero elastic tool HAWC [35].

In the first stage the focus has been in modelling of the induction machine and some special models including deep-bar effect, reduced order models (neglecting the stator transients) and steady state models have been developed. Next step is to extend the library with the corresponding models for the synchronous machine both field winding and permanent magnet.

Since all these electrical machines models involves a relatively big number of derivates for the state variables two methods have been chosen in order to implement the models in Simulink: using Simulink blocks and C S-Function. In §3.2 it will be shown how the implementation method of the dynamic equations affects the simulation speed.

In Fig. 3.4 is shown the electrical machine library based on C S-Functions. An example of a C S-Function for the dq-model of the induction machine is shown in Appendix A.

![Electric Machinery C S-Function based models](image)

Fig. 3.4. Electric machinery C S-Function based models.

However, contrary to C S-Function the implementation method using Simulink blocks is much more open and accessible for the user to “see” the model and eventually to make modifications on it.

### 3.1.3 Power Converters Library

The third library contains models for power converters based on switching functions. At the moment the following models are available: 3-phase diode bridge rectifier, voltage source converter (VCS), soft-starters and different modulation strategies for power converters, as shown in Fig. 3.5.
Using two VSC blocks connected via a DC-link circuit a back-to-back converter topology can be obtained.

The connection type of the soft-starter can be selected from the mask interface, e.g. star connection, delta or branch delta connection.

Two types of modulation strategy for the Voltage Source Converter are available, namely sinusoidal PWM and Space Vector PWM both average and switching models. The switching models include minimum pulse width, dead-time compensation and pulse dropping.

### 3.1.4 Common Blocks Library

This library contains models related with the grid connection of the wind turbines as shown in Fig. 3.6. Some relevant models from this library are:

- Voltage source with a harmonic content according with the EN61000-2-2 standard is included in this library;
- Transformer model, which takes the core geometry as well as the iron losses into account, has been developed. The model is implemented both using Simulink blocks and C S-Function;
- Distribution line model is based on the dynamic equations of the short line without distortion, which is a relatively simple model with good results in the case of a distribution line;
- Switch model (on-off) of the circuit breaker, which takes different opening time moments for poles into account, has been developed;
- Capacitor bank model.
### 3.1.5 Transformations Library

As in a dynamic model of a wind turbine there are several transformations for the signals, a special library, which contains the main transformations, has been implemented as shown in Fig. 3.7. The basic transformations are from \( abc \) to \( dqo \) arbitrary reference and the opposite one, as well as from a \( dq \) signal to the complex representation of it. In order to take the connection type into account, some transformations from a three-phase system in phase quantities to a three-phase one but in line-to-line quantities have been implemented. In order to find the magnitude and the phase for a three-phase system the corresponding block can be used.

Some of these models e.g. \( ABC \rightarrow Magnitude\&Phase \) uses special blocks from *DSP Blockset*.

### 3.1.6 Measurements Library

This library contains some special blocks as: calculation of the period for a sinusoidal variable, calculation of the grid angle using a Phase Locked Loop system, calculation of the
average wind for a given time interval, different modes of active and reactive power calculation (mono-phase, \(dq\)-reference frame, complex), as shown in Fig. 3.8.

![Fig. 3.8. Measurements Library.]

Some of these models e.g. Average Wind uses special blocks from DSP Blockset.

### 3.1.7 Control Library

This library contains models related with the control of the wind turbine system.

Since the main component of a control scheme is the PI Controller, a model with an anti wind-up structure has been implemented in Simulink. The Maximum Power Point Tracker (MPPT) block is based on a look-up table obtained from the wind turbine characteristics.

A simplified control for active and reactive for a doubly fed induction generator is developed. This control algorithm can be used with a reduced order model of the generator.

![Fig. 3.9. Content of the Control Blocks Library.]

This library will be extended with other types of controllers as for example the active stall controller and capacitor bank controller used in fixed-speed wind turbines.
3.2 Simulation speed in Matlab/Simulink

Using the available blocks from Matlab/Simulink three methods have been considered to investigate the simulation speed. Each method implements the same set of dynamic equations for the induction machine in different ways. The general structure for these models is shown in Fig. 3.10.

![Simulink model of induction machine in Simulink.](image)

The first two models, Model A and Model B, have the same structure as shown in Fig. 3.11.

![Implementation of induction machine model in dq-rotor reference frame used in Model A and Model B.](image)

The only difference between Model A and Model B consists in the implementation of the state-space equations with currents as state-variables.

Model A implements the dynamic equations using Simulink blocks as *Sum*, *Product*, *Gain*, *GoTo*, *From* and *Integrator* as shown in Fig. 3.12. The same equations are implemented in Model B by using few blocks, e.g. *Function* and *Integrator* blocks.

The third model - Model C – shows in Fig. 3.12 is based on an S-function written in “C” language according with the Simulink format for this type of function. The coordinate transformations $abc/dq$ and $dq/abc$ are also included in this function as well as the calculation of the reference frame angle. So, the S-function called “scim” implements five differential equations.
Fig. 3.12. Simulink implementation of dynamic equations for Model A.

Fig. 3.13. Simulink implementation of dynamic equations for Model B.

Fig. 3.14. Simulink implementation of dynamic equations for Model C.
Using Simulink Performance Tools Toolbox® (Profiler) the considered models have been compared regarding the simulation speed. The Simulink Profiler collects performance data while simulating a model and generates a simulation report. This report is useful to find out how much time Simulink spends on executing each block from a model and hence where to focus the model optimization efforts.

The simulation mechanism in Simulink involves several callbacks to some specific Simulink Functions at each time step. Fig. 3.15 shows the pseudo-code, which summarizes the execution of a dynamic system in Simulink.

```plaintext
Sim() 
ModelInitialize(). 
ModelExecute() 
    for t = tStart to tEnd 
    Output() 
    Update() 
    Integrate() 
        Compute states from deriv by repeatedly calling: 
        MinorOutput() 
        MinorDeriv() 
    EndIntegrate 
    Set time t = tNew. 
EndModelExecute 
ModelTerminate 
EndSim
```

Fig. 3.15. Pseudo-code for execution of a dynamic system model in Simulink

According to this conceptual model, Simulink executes a dynamic model by invoking the functions presented in Table 3.1 zero, one, or more times, depending on the function and the model.

The Profiler measures the time required to execute each invocation of these functions and generates a report at the end of the model that details how much time was spent in each function.

<table>
<thead>
<tr>
<th>Function</th>
<th>Purpose</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>sim</td>
<td>Simulate the model. This top-level function invokes the other functions required to simulate the model. The time spent in this function is the total time required to simulate the model.</td>
<td>System</td>
</tr>
<tr>
<td>ModelInitialize</td>
<td>Set up the model for simulation</td>
<td>System</td>
</tr>
<tr>
<td>ModelExecute</td>
<td>Execute the model by invoking the output, update, integrate, etc., functions for each block at each time step from the start to the end of simulation.</td>
<td>System</td>
</tr>
<tr>
<td>Output</td>
<td>Compute the outputs of a block at the current time step.</td>
<td>Block</td>
</tr>
<tr>
<td>Update</td>
<td>Update a block’s state at the current time step.</td>
<td>Block</td>
</tr>
<tr>
<td>Integrate</td>
<td>Compute a block’s continuous states by integrating the state derivatives at the current time step.</td>
<td>Block</td>
</tr>
<tr>
<td>MinorOutput</td>
<td>Compute a block’s output at a minor time step.</td>
<td>Block</td>
</tr>
<tr>
<td>MinorDeriv</td>
<td>Compute a block’s state derivatives at a minor time step.</td>
<td>Block</td>
</tr>
<tr>
<td>MinorZeroCrossings</td>
<td>Compute a block’s zero crossing values at a minor time step.</td>
<td>Block</td>
</tr>
<tr>
<td>ModelTerminate</td>
<td>Free memory and perform any other end-of-simulation cleanup.</td>
<td>System</td>
</tr>
</tbody>
</table>
The considered models have been tested using Simulink Profiler for a simulation time of 0.3 s with a simulation step of 1 ms. Ode5 fixed step solver has been used.

Table 3.2. Simulink Profiler Summary Report for the considered models.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Model A</th>
<th>Model B</th>
<th>Model C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total recorded time</td>
<td>13.35 sec</td>
<td>6.98 sec</td>
<td>3.02 sec</td>
</tr>
<tr>
<td>Number of block methods</td>
<td>71</td>
<td>40</td>
<td>15</td>
</tr>
<tr>
<td>Number of internal methods</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

The parameters illustrated in Table 3.2 are:

- **Total recorded time** – total time required to simulate the model using Simulink Profiler;
- **Number of block methods** – total number of invocations of block-level functions (e.g. Output);
- **Number of internal methods** – total number of invocations of system-level functions (e.g. ModelExecute).

The number of blocks used in a model has a big influence on the total simulation time as shown in Table 2.2. Model A, which uses much more blocks than Model B, has the biggest recorded time even if the number of Integrator blocks is the same.

Fig. 3.16 shows the simulation time in percent for functions invoked to simulate the entire model. The functions for the constitutive blocks from the models have been omitted. Function Integrate is the most time consuming procedure during the simulation around 70-80 % from the total simulation time. So, the total simulation time for a Simulink model is affected first by the number of blocks used in the model and then by the number of Integrator blocks.

![Fig. 3.16. Total time spent executing all invocations of specific functions for the considered models as a percentage of the total simulation time for Model A.](image-url)
A comparative study regarding execution time for a C S-function block and one Integrator block, which compute the rotor speed in this case, for the Model C is presented in Fig. 3.17.

It has been considered the average time (self-time) required to execute Update, Output and MinorDeriv functions for these blocks as a percent from the sum of the self-time for these functions for the Integrator block. It is obvious that the total time required for all invocations of functions related to the S-function block is smaller than the corresponding time for one Integrator block even if this S-function implements five differential equations.

Therefore, using a S-function which implements all the differential equations from the considered dynamic system, the simulation speed can be increased with at least a factor of two. Moreover, the numerical stability increases using an S-Function.

So, in modelling of large dynamic systems, which involves a big number of Integrator blocks it is desirable to use C S-Function concept in order to increase the simulation speed and improve the numerical stability.

3.3 Summary

In this chapter the structure of the new Matlab/Simulink Toolbox for wind turbine applications developed during the Simulation Platform Project is presented. The content of the main libraries is briefly presented and then some aspects regarding the simulation speed for the developed models are shown.
Chapter 4

Description of the models

In this chapter a detailed description of the models from Wind Turbine Blockset is presented. The mathematical descriptions for the models as well as their Simulink implementation are shown.

4.1 Mechanical component models

4.1.1 Wind model

The wind model has been developed at RISO National Laboratory based on the Kaimal spectra. The wind speed is calculated as an average value of the fixed-point wind speed over the whole rotor, and it takes the tower shadow and the rotational turbulences into account [71].

A main component in this model is the normally distributed white noise generator. Therefore, in order to obtain the same wind time series in all considered simulation tools used in the Simulation Platform some investigations have been done. It has been found that the built-in white noise generator from different simulation tools uses a different algorithm and thus a different wind time series is obtained. A new normally distributed white noise generator has been implemented using a ‘C’ S-Function based on the Ziggurat Algorithm developed by G. Marsaglia [45].

Currently, the wind model is available in two basic versions. The first one uses the normally distributed white noise generator from Matlab/Simulink, while the second one is based on the new developed normally distributed white noise generator. The general structure of these Simulink models is shown in Fig. 4.1.

![Simulink implementation of the wind model](image)

Fig. 4.1. Simulink implementation of the wind model.
The parameters defined in the block's mask are: rotor diameter of the wind turbine, average wind speed, length scale, turbulence intensity and sample time as shown in Fig. 4.2.

![Mask interface for wind model](image)

In order to validate the new white noise generator model some comparisons have been done. A wind time series for 3600 sec with 0.05 sec sample time, an average wind speed of 10 m/sec and 12% turbulence intensity has been generated for both models as shown in Fig. 4.3.

![Wind time series for the considered models](image)

The 20 bin-width histograms for these wind time series are presented in Fig. 4.4.
Fig. 4.4. Wind histogram for the considered models (ZA – Ziggurat Algorithm, SB – built-in Simulink Block).

Finally, the power spectra density for both wind time series has been calculated as shown in Fig. 4.5.

Even if the wind time series for the considered models are not identically as shown in Fig. 4.3 the power spectra density is approximately the same, which is an important issue.

### 4.1.2 Wind turbine rotor model

The aerodynamic model of the wind turbine rotor is based on the torque coefficient $C_Q$ or the power coefficient $C_P$ look-up table. The torque coefficient $C_Q$ is used to determine the aerodynamic torque directly by using:

$$T_a = 0.5 \pi R^3 \alpha^2 C_Q$$  \hspace{1cm} (4.1.1)
where: \( \rho \) is the air density, \( R \) is the blade radius, \( v_{\infty} \) is the wind speed and \( C_Q \) is the torque coefficient.

Alternatively the aerodynamic torque can be determined using the power coefficient \( C_P \) based on:

\[
T_{\text{wt}} = 0.5\pi\rho R^2 v_{\infty}^3 C_P
\]

Important to underline that both coefficients \( C_P \) and \( C_Q \) can be function of the tip speed ratio \( \lambda \) for passive-stall wind turbines or function of tip speed ratio \( \lambda \) and pitch angle \( \theta \) for active stall and variable pitch/speed wind turbines.

The Simulink model for the variable pitch wind turbine rotor is presented in Fig. 4.6.

![Simulink model of the wind turbine rotor.](image)

The parameters from the mask interface are: blade radius, air density, cut-in and cut-out wind speeds, as shown in Fig. 4.7.

![Mask interface for the wind model in Simulink.](image)

Using a reduced look-up table in respect with the variation of the torque coefficient/power coefficient with the pitch angle \( (-10^\circ \leq \theta \leq 10^\circ) \) the model for the active-stall wind turbine is obtained. Imposing a zero value for the pitch angle \( \theta \) the model for passive stall wind turbine can be obtained.
4.1.3 Drive trains models

Starting from the three-mass model for a wind turbine drive train the two-mass model and the one-mass model are derived.

4.1.3.1 Three-mass model

The equivalent model of a wind turbine drive train is presented in Fig. 4.8. The masses correspond to a large mass of the wind turbine rotor, masses for the gearbox wheels and a mass for generator respectively.

![Diagram of wind turbine drive train](image)

Fig. 4.8. Two-mass model of a wind turbine drive train.

Taking into account the stiffness and the damping factors for both shafts the dynamic equations can be written as [66]:

\[
\begin{align*}
T_{\text{wtr}} &= J_{\text{wtr}} \frac{d\Omega_{\text{wtr}}}{dt} + D_{\text{wtr}} \Omega_{\text{wtr}} + k_{\text{swtr}} (\theta_{\text{wtr}} - \theta_1) \\
T_1 &= J_1 \frac{d\Omega_1}{dt} + D_{\text{wtr}} \Omega_1 + k_{\text{swtr}} (\theta_1 - \theta_{\text{wtr}}) \\
T_2 &= J_2 \frac{d\Omega_2}{dt} + D_{\text{gen}} \frac{d\Omega_2}{dt} + k_{\text{sgen}} (\theta_2 - \theta_{\text{gen}}) \\
-T_{\text{gen}} &= J_{\text{gen}} \frac{d\Omega_{\text{gen}}}{dt} + D_{\text{gen}} \Omega_{\text{gen}} + k_{\text{sgen}} (\theta_{\text{gen}} - \theta_2)
\end{align*}
\]

where: 
- \( T_{\text{wtr}} \) - wind turbine torque; 
- \( J_{\text{wtr}} \) - wind turbine moment of inertia; 
- \( \Omega_{\text{wtr}} \) - wind turbine mechanical speed; 
- \( k_{\text{swtr}} \) - spring constant indicating the torsional stiffness of the shaft on wind turbine part; 
- \( T_{\text{gen}} \) - generator torque; 
- \( J_{\text{gen}} \) - generator moment of inertia; 
- \( \Omega_{\text{gen}} \) - generator mechanical speed; 
- \( k_{\text{sgen}} \) - spring constant indicating the torsional stiffness of the shaft on generator part; 
- \( T_1 \) - torque that goes in the gearbox; 
- \( T_2 \) - torque out from the gearbox; 
- \( T_2 = -\frac{1}{K_{\text{gear}}} \cdot T_1 \) - gearbox ratio; 
- \( \Omega_2 = k_{\text{gear}} \cdot \Omega_1 \).
4.1.3.2 Two-mass model

The three-mass model can be reduced to a two-mass model by considering an equivalent system with an equivalent stiffness and damping factor. The moment of inertia for the shafts and the gearbox wheels can be neglected because they are small compared with the moment of inertia of the wind turbine or generator. Therefore the resultant model is essentially a two mass model connected by a flexible shaft. Only the gearbox ratio has influence on the new equivalent system.

The dynamic equations can be written in two points: on the wind turbine side with the influence of generator component through the gearbox and on the generator side respectively. The equivalent system on the generator side is shown in Fig. 4.9.

\[
\begin{align*}
\tau_{\text{wtr}}' &= J_{\text{wtr}} \frac{d\Omega_{\text{wtr}}'}{dt} + D_c' (\Omega_{\text{wtr}}' - \Omega_{\text{gen}}') + k_{\text{sc}}' (\theta_{\text{wtr}}' - \theta_{\text{gen}}') \\
\frac{d\theta_{\text{wtr}}'}{dt} &= \Omega_{\text{wtr}}' \\
-\tau_{\text{gen}}' &= J_{\text{gen}} \frac{d\Omega_{\text{gen}}'}{dt} + D_c' (\Omega_{\text{gen}}' - \Omega_{\text{wtr}}') + k_{\text{sc}}' (\theta_{\text{gen}}' - \theta_{\text{wtr}}') \\
\frac{d\theta_{\text{gen}}'}{dt} &= \Omega_{\text{gen}}'
\end{align*}
\]

where: the equivalent stiffness is given by:

\[
\frac{1}{k_{\text{sc}}'} = \frac{1}{k_{\text{wtr}}} + \frac{1}{k_{\text{gen}}} + \frac{1}{k_{\text{gear}}^2}
\]

and the equivalent moment of inertia for the rotor is:

\[
J_{\text{wtr}}' = \frac{1}{k_{\text{gear}}^2} \cdot J_{\text{wtr}}
\]

The Simulink implementation of (4.1.4) is shown in Fig. 4.10.
The parameters from the mask interface are: moment of inertia for machine and wind turbine rotor, equivalent stiffness and damping coefficients for the shafts, gearbox ratio and initial conditions for the state variables (speeds and angles for the low speed and high speed shafts), as shown in Fig. 4.11.

4.1.3.3 One-mass model

When the stiffness and the damping factor are neglected further reduction of the drive train model can be obtained. The one-mass model is obtained in this case, which is described by:

\[ T_{\text{gen}} - T_{\text{wt}} = J_{\text{gen}} \frac{d\Omega_{\text{gen}}}{dt} \]  

\( (4.1.7) \)
where: the equivalent moment of inertia is \( J_{\text{ech}} = J_{\text{gen}} + \frac{J_{\text{wr}}}{k_{\text{wr}}^2} \) and the equivalent wind turbine torque is \( T_{\text{wr}} = \frac{T_{\text{wr}}}{k_{\text{wr}}^2} \).

The Simulink implementation of (4.1.7) is presented in Fig. 4.12.

**4.2 Electrical machinery models**

In this paragraph the dynamic equations and the Simulink implementation for the generators used in wind turbine applications are shown.

**4.2.1 Induction machine models**

In this paragraph the dynamic equations of induction machine in the natural reference frame \( abc \) as well as in the \( d\alpha\) arbitrary reference frame are presented [34]. These equations are written using both fluxes and currents as state-variables.

**4.2.1.1 ABC/\( abc \) model of induction machine**

The basic equations of the machine (assuming sinusoidal \( mmf \) and neglecting saturation and losses in the core) by using the diagram shown in Fig. 4.13 are [9], [23], [34], [54]:

![Simplified diagram of induction machine with rotor and stator windings.](image-url)
\begin{align}
\mathbf{V} &= \mathbf{R} \mathbf{I} + \frac{d\mathbf{\Psi}}{dt} \\
\mathbf{\Psi} &= \mathbf{L} \mathbf{I}
\end{align} \quad (4.2.1)

where:

- \( \mathbf{V} = [v_A, v_B, v_C, v_s, v_b, v_c] \) - are the voltages applied to each stator and rotor phases;
- \( \mathbf{I} = [i_A, i_B, i_C, i_s, i_b, i_c] \) - are the currents in each stator and rotor phases;
- \( \mathbf{\Psi} = [\phi_A, \phi_B, \phi_C, \phi_s, \phi_b, \phi_c] \) - are the fluxes linked with each stator and rotor phases;
- \( \mathbf{R} \) is the resistance matrix and \( \mathbf{L} \) is the inductance matrix.

Substituting (4.2.2) in (4.2.1) and arrange the equation in state-variable form results in:

\begin{equation}
\frac{d\mathbf{I}}{dt} = \mathbf{L}^{-1} \left( -\mathbf{R} - \frac{d\mathbf{L}}{dt} \right) \mathbf{I} + \mathbf{L}^{-1} \mathbf{V} \quad (4.2.3)
\end{equation}

Most terms of the inductance matrix \( \mathbf{L} \) are functions of the rotor position \( \theta_r \). The derivative of the inductance matrix from (4.2.3) can be written as:

\begin{equation}
\frac{d\mathbf{L}}{dt} = \frac{d\mathbf{L}}{d\theta_r} \frac{d\theta_r}{dt} = \frac{d\mathbf{L}}{d\theta_r} \omega_r \quad (4.2.4)
\end{equation}

where \( \omega_r \) is the rotor speed. Grouping (4.2.3) and (4.2.4) results in:

\begin{equation}
\frac{d\mathbf{I}}{dt} = \mathbf{L}^{-1} \left( -\mathbf{R} - \omega_r \frac{d\mathbf{L}}{d\theta_r} \right) \mathbf{I} + \mathbf{L}^{-1} \mathbf{V} \quad (4.2.5)
\end{equation}

which is the state space form of the dynamic equations for the induction machine.

The inductance matrix is defined as:

\begin{equation}
\mathbf{L} = \begin{bmatrix}
L_s & -0.5M_s & -0.5M_s & M_{sr f_1} & M_{sr f_2} & M_{sr f_3} \\
-0.5M_s & L_s & -0.5M_s & M_{sr f_1} & M_{sr f_2} & M_{sr f_3} \\
-0.5M_s & -0.5M_s & L_s & M_{sr f_3} & M_{sr f_1} & M_{sr f_2} \\
M_{sr f_1} & M_{sr f_2} & M_{sr f_3} & L_r & -0.5M_r & -0.5M_r \\
M_{sr f_2} & M_{sr f_1} & M_{sr f_3} & -0.5M_r & L_r & -0.5M_r \\
M_{sr f_3} & M_{sr f_2} & M_{sr f_1} & -0.5M_r & -0.5M_r & L_r
\end{bmatrix} \quad (4.2.6)
\end{equation}

where:

- \( L_s = L_{os} + M_s \) - stator self-inductance;
- \( L_r = L_{or} + 1.5M_r \) - rotor self-inductance;
- \( L_{os}, L_{or} \) - stator and rotor leakage inductance;
- \( M_s, M_r \) - stator and rotor mutual inductance;
- \( M_{sr} \) - stator-rotor mutual inductance.
The coefficients $f_1, f_2, f_3$ are:

$$f_1 = \cos \theta_r, \quad f_2 = \cos \left( \theta_r + \frac{2\pi}{3} \right), \quad f_3 = \cos \left( \theta_r - \frac{2\pi}{3} \right) \quad (4.2.7)$$

All these inductances can easily be measured in a wound rotor induction machine. For squirrel-cage induction machines, these inductances can normally be estimated from no-load and locked rotor tests. Referring the rotor parameters to stator the mutual inductances become equals:

$$M_s = M_r = M_{sr} \quad (4.2.8)$$

If it is taken into account that the machines windings forms an balanced/symmetrical system and:

$$i_A + i_B + i_C = 0 \quad \text{and} \quad i_a + i_b + i_c = 0 \quad (4.2.9)$$

the inductance matrix $[L]$ can be rewritten as:

$$[L] = \begin{bmatrix}
0 & 0 & M_{sr}f_1 & M_{sr}f_2 & M_{sr}f_3 \\
0 & 0 & M_{sr}f_3 & M_{sr}f_1 & M_{sr}f_2 \\
M_{sr}f_1 & M_{sr}f_3 & M_{sr}f_2 & 0 & 0 \\
M_{sr}f_2 & M_{sr}f_1 & M_{sr}f_3 & 0 & 0 \\
M_{sr}f_3 & M_{sr}f_2 & M_{sr}f_1 & 0 & 0 \end{bmatrix} \quad (4.2.10)$$

where: $L_s = L_{os} + 1.5M_s$ and $L_r = L_{or} + 1.5M_s$.

From (4.2.10) the results are:

$$\frac{d[L]}{d\theta_r} = -M_{sr} \begin{bmatrix}
0 & 0 & g_1 & g_2 & g_3 \\
0 & 0 & g_3 & g_1 & g_2 \\
0 & 0 & g_2 & g_3 & g_1 \\
g_1 & g_3 & g_2 & 0 & 0 \\
g_3 & g_2 & g_1 & 0 & 0 \end{bmatrix} \quad (4.2.11)$$

and

$$g_1 = \sin \theta_r, \quad g_2 = \sin \left( \theta_r + \frac{2\pi}{3} \right), \quad g_3 = \sin \left( \theta_r - \frac{2\pi}{3} \right) \quad (4.2.12)$$

The main problem in implementing (4.2.5) is to calculate the inverse of the inductance matrix at each simulation step [23]. A solution is to use modern software tools, which have these capabilities (e.g. Matlab). However, an analytical expression for this matrix is useful since the coefficients depend on the electrical angle $\theta_r$. One approach is to apply a Cholesky decomposition since the inductance matrix is “symmetric positive definite”[23].

Starting from the general form (4.2.10) the inverse of the inductance matrix can be written as:
\[ [L]^\dagger = [B] = \begin{bmatrix}
  b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \\
  b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} \\
  b_{31} & b_{32} & b_{33} & b_{34} & b_{35} & b_{36} \\
  b_{41} & b_{42} & b_{43} & b_{44} & b_{45} & b_{46} \\
  b_{51} & b_{52} & b_{53} & b_{54} & b_{55} & b_{56} \\
  b_{61} & b_{62} & b_{63} & b_{64} & b_{65} & b_{66}
\end{bmatrix} \]  

(4.2.13)

Using a symbolic mathematical tool e.g. Matlab, Maple or Mathcad the coefficients from (4.2.13) can be determined easily. First, some coefficients are calculated as:

\[ K_1 = \frac{L_r L_s}{M_w}, \quad K_2 = \frac{3}{K_1 - \frac{4}{9}}, \quad K_3 = \frac{4}{9}, \quad K_4 = \frac{9}{K_1 - \frac{4}{9}} \]  

(4.2.14)

and then all other elements of the matrix B:

\[ b_{11} = b_{22} = b_{33} = \frac{K_2}{L_s}, \quad b_{44} = b_{55} = b_{66} = \frac{K_3}{L_r} \]

(4.2.15)

The electromagnetic torque can be obtained starting from the voltage equations using currents as state variables

\[ [V] = [R][I] + \frac{d}{dt} \left([L][I]\right) = [R][I] + \frac{d[L]}{dt}[I] + [L]\frac{d[I]}{dt} \]  

(4.2.16)

and multiplying it with the transpose of the currents matrix:

\[ [I]^\dagger [V] = [I]^\dagger [R][I] + [I]^\dagger \left(\frac{d}{dt}[\Psi]\right) = [I]^\dagger [R][I] + [I]^\dagger \frac{d[L]}{dt}[I] + [I]^\dagger [L]\frac{d[I]}{dt} \]  

(4.2.17)

Alternatively using

\[ \frac{d[L]}{dt} = \frac{d[L]}{d\theta_r} \cdot \frac{d\theta_r}{dt} = \frac{d[L]}{d\theta_r} \cdot \omega_r, \]  

where \( \omega_r \) is the electrical speed of the rotor results in:

\[ [I]^\dagger [V] = [I]^\dagger [R][I] + \omega_r[I]^\dagger \frac{d[L]}{d\theta_r}[I] + [I]^\dagger [L]\frac{d[I]}{dt} \]  

(4.2.18)

The terms from (4.2.18) have the following meaning:
P; = \begin{bmatrix} I \end{bmatrix} \begin{bmatrix} V \end{bmatrix} \quad \text{- instantaneous power;}

P_{\text{copper}} = \begin{bmatrix} I \end{bmatrix} \begin{bmatrix} R \end{bmatrix} \begin{bmatrix} I \end{bmatrix} \quad \text{- copper losses in the machine windings;}

P_{\text{mag}} = \begin{bmatrix} I \end{bmatrix} \begin{bmatrix} L \end{bmatrix} \frac{d\begin{bmatrix} I \end{bmatrix}}{dt} \quad \text{- magnetic power stored in the machine (due to the variation in time of the magnetic energy);}

P_m = \omega_r \frac{d\begin{bmatrix} L \end{bmatrix}}{d\theta_r} \begin{bmatrix} I \end{bmatrix} \quad \text{- mechanical power.}

The electromagnetic torque is then:

\[ T_e = \frac{P_m}{\Omega_r} = \frac{P_m}{\omega_r} = p \frac{\begin{bmatrix} I \end{bmatrix}}{\begin{bmatrix} I \end{bmatrix}} \frac{d\begin{bmatrix} L \end{bmatrix}}{d\theta_r} \begin{bmatrix} I \end{bmatrix} \quad (4.2.19) \]

where: \( p \) is the number of pole pairs and \( \Omega_r \) is the mechanical speed of the rotor. Substituting \((4.2.11)\) in \((4.2.19)\) the electromagnetic torque as a function of currents is:

\[ T_e = -pM_{sr} \{ (i_Ai_A + i_Bi_B + i_ci_c) \sin \theta + (i_Ai_B + i_Bi_A + i_ci_i_a) \sin(\theta + 2\pi/3) \\
+ (i_Ai_c + i_Bi_a + i_ci_i_b) \sin(\theta - 2\pi/3) \} \quad (4.2.20) \]

Based on \((4.2.5)\) and \((4.2.20)\) the dynamic model of the induction machine can easily be implemented in Matlab/Simulink, as shown in Fig. 4.14.

\[ \text{Fig. 4.14. Simulink implementation of the ABC/abc model for a squirrel-cage induction machine.} \]

In Fig. 4.15 the mask interface for this model is shown. The input parameters in this interface are based on the standard data sheet: resistance and leakage inductance for the stator and rotor.
windings, magnetizing inductance, number of pole pair and initial conditions for the state-variables (currents and electrical angle of the rotor). Moreover, the operating mode, motor or generator, as well as the connection type for the stator windings, star or delta, can be selected.

Block Parameters: ABC/abc Squirrel-Cage

ABC/abc model of Squirrel Cage Induction Machine (mask)

Implements an ABC/abc model for squirrel-cage induction machine. Currents as state variable.

- Operating mode: Generator
- Connection type: Star
- Stator windings: [Rs Lsgm_s] [ohms H]
  - [0.005 4.0744e-004j]
- Rotor windings: [Rr Lsgm_r] [ohms H]
  - [0.0089 2.9921e-004]
- Magnetizing inductance: 0.0160 H
- Number of poles pair: 2
- Initial condition: [iA iB iC ia ib ic theta]
  - [0 0 0 0 0 0 0]

A similar model and mask interface are available for wound rotor induction machine

### 4.2.1.2 Model of induction machine in dqo arbitrary reference frame

Some of the machine inductances are functions of the rotor speed, whereupon the coefficients of the state-space equations (voltage equations), which describe the behaviour of the induction machine, are time-varying (except when the rotor is at stand-still). A change of variables is often used to reduce the complexity of these state-space equations. There are several changes of variables, which are used but there is just one general transformation [34]. This general transformation refers the machine variable to a frame of reference, which rotates at an arbitrary angular velocity \( \omega_g \). In this reference frame the machine windings are replaced with some equivalent windings as shown in Fig. 4.16.

![Induction machine windings in the dqo-arbitrary reference frame.](image)

Based on (4.2.1) and using the general transformation, the voltage equations in the arbitrary reference frame can be written as:
\[
\begin{bmatrix}
    v_{sd} \\
v_{sq} \\
v_{so} \\
v_{rd} \\
v_{rq} \\
v_{ro}
\end{bmatrix} =
\begin{bmatrix}
    R_s & 0 & 0 & i_{sd} & i_{sq} & i_{so} \\
    R_s & 0 & 0 & i_{rd} & i_{rq} & i_{ro} \\
    0 & R_r & 0 & i_{rd} & i_{rq} & i_{ro} \\
    0 & R_r & 0 & i_{rd} & i_{rq} & i_{ro}
\end{bmatrix}
+ \frac{d}{dt}\begin{bmatrix}
    \Psi_{sd} & \Psi_{sq} & \Psi_{so} \\
    \Psi_{rd} & \Psi_{rq} & \Psi_{ro}
\end{bmatrix}
+ \begin{bmatrix}
    0 & -\omega_s & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & \omega_s & 0 & 0 \\
    0 & 0 & 0 & 0 & 0 & \omega_s \\
    0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
    \Psi_{sd} \\
    \Psi_{sq} \\
    \Psi_{so} \\
    \Psi_{rd} \\
    \Psi_{rq} \\
    \Psi_{ro}
\end{bmatrix}
\]

(4.2.21)

where: \(\omega_r\) is the electrical speed of the machine and \(\omega_g\) is the speed of the general reference frame.

The compact form of (4.2.21) is:

\[
[V] = [R][I] + \frac{d}{dt}[\Psi] + [\Omega][\Psi]
\]

(4.2.22)

The relation between the linkage fluxes and the currents is given by:

\[
\begin{bmatrix}
    \Psi_{sd} \\
    \Psi_{sq} \\
    \Psi_{so} \\
    \Psi_{rd} \\
    \Psi_{rq} \\
    \Psi_{ro}
\end{bmatrix} =
\begin{bmatrix}
    L_s & 0 & 0 & L_m & 0 & 0 \\
    0 & L_s & 0 & 0 & L_m & 0 \\
    0 & 0 & L_{eS} & 0 & 0 & i_{so} \\
    L_m & 0 & 0 & L_r & 0 & 0 \\
    0 & L_m & 0 & 0 & L_r & 0 \\
    0 & 0 & 0 & 0 & L_{eR} & i_{ro}
\end{bmatrix}
\begin{bmatrix}
    i_{sd} \\
    i_{sq} \\
    i_{so} \\
    i_{rd} \\
    i_{rq} \\
    i_{ro}
\end{bmatrix}
\]

(4.2.23)

or in compact form:

\[
[\psi] = [L][I]
\]

(4.2.24)

where:

- \(L_m = \frac{3}{2} M_{sr}\) is the mutual inductance in the general reference frame;
- \(L_s = L_{eS} + L_m\) is the stator self inductance;
- \(L_r = L_{eR} + L_m\) is the rotor self inductance.

The voltage equations are written again in terms of currents and flux linkages. Clearly, these variables are related based on the matrix inductance \([L]\) and both cannot be independent or state variables.

**Fluxes as state variables**

If the fluxes are selected as state variables the currents can be expressed in compact form by:

\[
[I] = [L]^{-1}[\psi]
\]

(4.2.25)

The inverse of the inductance matrix is:
Wind Turbine Blockset in Matlab Simulink

\[
[L]^{-1} = \frac{1}{D} \begin{bmatrix}
0 & 0 & -L_m & 0 & 0 \\
0 & L_r & 0 & 0 & -L_m & 0 \\
0 & 0 & -\frac{D}{L_{oT}} & 0 & 0 & 0 \\
-L_m & 0 & 0 & L_s & 0 & 0 \\
0 & -L_m & 0 & 0 & L_s & 0 \\
0 & 0 & 0 & 0 & -\frac{D}{L_{oT}} & 0
\end{bmatrix}
\]

where \( D = L_r L_s - L_m^2 \).

After some mathematical manipulation, the state-space form of the dynamic equations (4.2.21) is:

\[
\frac{d}{dt}[[\Psi]] = -[[R][L]^{-1}][\Psi] + [V]
\]

or in an expanded form:

\[
\begin{bmatrix}
\frac{d}{dt} \Psi_{sd} \\
\frac{d}{dt} \Psi_{sq} \\
\frac{d}{dt} \Psi_{so} \\
\frac{d}{dt} \Psi_{rd} \\
\frac{d}{dt} \Psi_{rq}
\end{bmatrix} =
\begin{bmatrix}
\frac{R L_s}{D} & -\omega_g & 0 & -\frac{R L_m}{D} & 0 \\
0 & \frac{R L_s}{D} & 0 & 0 & -\frac{R L_m}{D} & 0 \\
-\frac{R L_m}{D} & 0 & 0 & \frac{R L_s}{D} & -\left(\omega_g - \omega_r\right) & 0 \\
0 & -\frac{R L_m}{D} & 0 & \left(\omega_g - \omega_r\right) & \frac{R L_s}{D} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{R r}{L_{oT}}
\end{bmatrix}
\begin{bmatrix}
\Psi_{sd} \\
\Psi_{sq} \\
\Psi_{so} \\
\Psi_{rd} \\
\Psi_{rq}
\end{bmatrix} +
\begin{bmatrix}
v_{sd} \\
v_{sq} \\
v_{so} \\
v_{rd} \\
v_{rq}
\end{bmatrix}
\]

Currents as state variables

If the currents are selected as state variables the fluxes will be replaced based on (4.2.23). Assuming that all parameters are constant (saturation and other effects are not taken into account) (4.2.21) become:

\[
\frac{d}[I]\frac{dt}{dt} = -([L]^{-1}[[R]] + [L]^{-1}[[\Omega][L]][I] + [L]^{-1}[[V]]
\]

So the state-space form of the dynamic equations using currents as state variables is:

\[
\frac{d}[I]\frac{dt}{dt} = [A][I] + [B][V]
\]

where:
The electromagnetic torque can be obtained starting from (4.2.22) and multiplying it from the left with the transpose of the currents vector:

\[ [A] = \begin{bmatrix}
\frac{R}{L} & \frac{L}{D} & 0 & \frac{R}{L_m} & \frac{L}{D} \\
\frac{L}{D} & 0 & 0 & \frac{R}{L_m} & 0 \\
0 & 0 & 0 & \frac{R D}{L_m} & 0 \\
\frac{R}{L_m} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \]

\[ [B] = [L]^{-1} = \frac{1}{D} \begin{bmatrix}
L & 0 & 0 & -L_m & 0 & 0 \\
0 & L & 0 & 0 & -L_m & 0 \\
0 & 0 & \frac{D}{L_{em}} & 0 & 0 & 0 \\
-L_m & 0 & 0 & L & 0 & 0 \\
0 & -L_m & 0 & 0 & L & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{D}{L_{em}} \\
\end{bmatrix} \]

\[ \mathbf{\text{Electromagnetic torque}} \]

The electromagnetic torque can be obtained starting from (4.2.22) and multiplying it from the left with the transpose of the currents vector:

\[ [I] [V] = [I] [R][I] + [I] \frac{d}{dt} [\Psi] + [I] [\Omega][\Psi] \quad (4.2.31) \]

where

- \( P_i = [I] [V] \) is the instantaneous power;
- \( P_{\text{copper}} = [I] [R][I] \) are the copper losses in the machine windings;
- \( P_{\text{mag}} = [I] \frac{d}{dt} [\Psi] \) is the magnetic power stored in machine (due to the variation in time of the magnetic energy);
- \( P_m = [I] [\Omega][\Psi] \) is the mechanical power.

The electromagnetic torque is then:

\[ T_e = \frac{p_m}{\Omega_r} = p \frac{p_m}{\omega_r} = \frac{3}{2} p \left( \Psi_{sd} i_{sq} - \Psi_{sq} i_{sd} \right) \quad (4.2.32) \]

where: \( p \) is the number of pole pairs and \( \Omega_r \) is the mechanical speed of the rotor.

Other equivalent expressions for the electromagnetic torque of an induction machine are:
Wind Turbine Blockset in Matlab Simulink

- \( T_e = \frac{3}{2} p \left( \Psi_{rd} i_{rd} - \Psi_{rd} i_{rq} \right) \)
- \( T_e = \frac{3}{2} p L_m \left( i_{sq} i_{rd} - i_{ad} i_{rq} \right) \)
- \( T_e = \frac{3}{2} p \frac{L_m}{D} \left( \Psi_{sq} \Psi_{rd} - \Psi_{ad} \Psi_{rq} \right) \)

The above equations may be somewhat misleading since they seem to imply that the leakage inductances are involved in the energy conversion process. This, however, is not the case. Even though the flux linkages contain the leakage inductances, they are eliminated by the algebra.

The Simulink implementation and the mask interface for this model have the same structure as for the abc/abc model (see Fig. 4.14 and Fig. 4.15).

4.2.1.3 Reduced order model

The theory of neglecting electric transients is presented in [34]. For balanced steady-state operation, the variables in synchronous reference frame are constants. Hence, the electric transients of the stator can be neglected by setting the derivative of the stator fluxes in the \( d \) and \( q \) axis to zero in [34] as:

\[
\begin{bmatrix}
V_{sd} \\
V_{sq} \\
V_{rd} \\
V_{rq}
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\psi_{sd} \\
\psi_{sq} \\
\psi_{rd} \\
\psi_{rq}
\end{bmatrix} + \begin{bmatrix}
\frac{R_s L_m}{D} & -\omega_e & -\frac{R_s L_m}{D} & 0 \\
\omega_e & \frac{R_L}{D} & 0 & -\frac{R_L}{D} \\
-\frac{R_L}{D} & 0 & \frac{R_L}{D} & -(\omega_e - \omega_r) \\
0 & \frac{R_L}{D} & 0 & \frac{R_L}{D}
\end{bmatrix}
\begin{bmatrix}
\psi_{sd} \\
\psi_{sq} \\
\psi_{rd} \\
\psi_{rq}
\end{bmatrix}
\]

In order to avoid the computational problems due to algebraic loops (4.2.33) should be written in state space form only in terms of the rotor fluxes as:

\[
\begin{bmatrix}
\psi_{rd} \\
\psi_{rq}
\end{bmatrix} =
\begin{bmatrix}
\frac{\alpha \tau_m}{\tau_r} \sigma_r - \frac{\tau_m}{\sigma_s} \\
-\left( \frac{\alpha \tau_m}{\tau_r} \frac{L_m + 1}{\tau_r} \omega_e - \omega_r \right) \alpha \tau_m \sigma_r - \frac{\tau_m}{\sigma_s}
\end{bmatrix}
\begin{bmatrix}
\psi_{rd} \\
\psi_{rq}
\end{bmatrix} + \begin{bmatrix}
\frac{\alpha \tau_m}{\tau_r} & \frac{\alpha \tau_m}{\tau_r} & 0 \\
\frac{\alpha \tau_m}{\tau_r} & \frac{\alpha \tau_m}{\tau_r} & 0
\end{bmatrix}
\begin{bmatrix}
V_{sd} \\
V_{sq}
\end{bmatrix}
\]

where: \( \tau_m = \frac{R_s L_m}{D} \), \( \tau_r = \frac{R_L}{D} \), \( \sigma_r = \frac{L_m}{L_{rr}} \), \( \sigma_s = \frac{L_m}{L_{ss}} \), \( \alpha = \frac{1}{\omega_e + \left( \frac{\omega_e}{\tau_r} \right)^2} \).

Based on rotor fluxes and input voltages the stator fluxes can be obtained using:
Then the electromagnetic torque is calculated based on:

\[
T_e = \frac{3}{2} \frac{L_m}{D} \left( \psi_{sq} \cdot \psi_{rd} - \psi_{rq} \cdot \psi_{sd} \right)
\]

However, the reduced order model, which neglects the stator transients, can be extracted from the full order model given by (4.2.30) written in synchronous reference frame using Control System Toolbox from Matlab. This toolbox is a very powerful tool for reduced order modeling. Based on some specific functions as `modred`, `ssbal`, `minreal`, and `balreal` the desired state variables can be eliminated from the state space equations and the reduced order models are obtained.

The Simulink model for this machine is based on (4.2.34)-(4.2.36) as shown in Fig. 4.17.

The mask interface for this model has the same structure as for the abc/abc model (see Fig. 4.15). The only difference consists in the initial conditions for the state variables, which are the fluxes in this case.

### 4.2.1.4 Steady-state model

The voltage equations, which describe the balanced steady-state operation of an induction machine, can be obtained in several ways [34]. For balanced steady state conditions the \(d\) and \(q\) variables are sinusoidal in all reference frames except the synchronously rotating reference frame wherein they are constants. In the following paragraphs the prime index will be used for the rotor equations in order to highlight that these are related to the stator circuit of the machine.
Starting from (4.2.21) written in the synchronously rotating reference frame $\omega_s = \omega_e$ and neglecting the time rate changing of all flux linkages and then employ the relationships:

$$\sqrt{2} F_{sd} = F_{sd} - jF_{sq}$$
$$\sqrt{2} F_{sr} = F_{rd} - jF_{rq}$$

the steady-state voltage equations for induction machine can be written as follows:

$$\overline{V}_s = (R_s + jX_{ss})\overline{I}_s + jX_m (\overline{I}_s + \overline{I}_r)$$
$$\overline{V}_r = (R_r + jsX_{sr})\overline{I}_r + jsX_m (\overline{I}_s + \overline{I}_r)$$

where: slip $s = \frac{\omega_e - \omega_s}{\omega_e} = \frac{2\pi f_s - p\Omega_r}{2\pi f_s} = 1 - \frac{p\Omega_r}{2\pi f_s}$; $f_s$ is the frequency of stator voltage; $\Omega_r$ is the mechanical rotor speed; $p$ is the number of pole pair.

Equation (4.2.38) suggests the per-phase equivalent circuit shown in Fig. 4.18.

![Fig. 4.18. Equivalent circuit for steady state operation of a symmetrical induction machine.](image)

The steady-state voltage equations (4.2.38) can be arranged as follows:

$$\overline{V}_s = (R_s + jX_s)\overline{I}_s + jX_m (\overline{I}_s + \overline{I}_r)$$
$$\overline{V}_r = (R_r + jsX_r)\overline{I}_r + jsX_m (\overline{I}_s + \overline{I}_r)$$

where:

$$X_s = X_{ss} + X_m$$ is the self reactance of the stator windings;

$$X_r = X_{sr} + X_m$$ is the self reactance of the rotor windings.

Or in matrix form:

$$\begin{bmatrix} \overline{V}_s \\ \overline{V}_r \end{bmatrix} = \begin{bmatrix} R_s + jX_s & jX_m \\ jsX_m & R_r + jsX_r \end{bmatrix} \begin{bmatrix} \overline{I}_s \\ \overline{I}_r \end{bmatrix}$$

The compact form of (4.2.40) is:

$$\overline{V} = [\overline{Z}] [\overline{I}]$$
Based on (4.2.41) the stator and rotor currents are obtained from:

\[
[I] = [Z]^{-1} \cdot [V]
\]  

(4.2.42)

So, the induction machine can be treated as a complex quadri-pole as shown in Fig. 4.19 where the complex impedance is a function of the slip.

Recalling (4.2.31) the electromagnetic torque is given by:

\[
T_e = \frac{3}{2} p X_m \text{Re} \left[ j \bar{l}_r \bar{l}_s \right]
\]  

(4.2.43)

where \( \bar{l}_s \) is the conjugate of \( l_s \).

The active power is given by:

\[
P_x = \frac{3}{2} \cdot \text{Re} \left( V_x \cdot \bar{l}_s \right)
\]  

(4.2.44)

The reactive power is given by:

\[
Q_x = \frac{3}{2} \cdot \text{Im} \left( V_x \cdot \bar{l}_s \right)
\]  

(4.2.45)

These equations can be used both for stator and rotor windings by replacing the indices \( x \) with \( s \) and \( r \) respectively.

Using this approach the induction machine can be completely characterized in terms of steady-state values of the stator and the rotor currents, phase angle for both currents, electromagnetic torque, active and reactive power for both stator and rotor circuit. Moreover, an analysis, both in motor and generator mode of operation, can be performed.

The Simulink implementation of the steady-state model, both for a squirrel-cage machine and wound rotor one, is shown in Fig. 4.20.

This model uses some particular blocks from the DSP Blockset (Simulink) such as: Matrix Multiplication, Horizontal and Vertical Concatenation, and LU Inverse (Matrix inverse using...
LU factorization). It has been chosen to use these blocks because the model also implements the rotor deep-bar effect. Using these blocks the model can be implemented in matrix form with the minimal number of blocks. However, the model can also be implemented using standard blocks from Simulink.

4.2.1.5 Modelling deep-bar effect

In larger machines like MW generators rotor deep-bar effect causes the equivalent rotor circuit resistance and leakage reactance to vary significantly with the slip [3], [46]. Fig. 4.21 illustrates the leakage flux paths for a squirrel-cage induction machine with deep-bars. The parts of each bar extending deeply into the rotor iron have higher leakage inductances than those parts of the bar cross-section near the air gap, because more of the leakage flux links the deeper parts of the bar. One may think of each bar as being composed of several layers of equal cross section, and thus of equal resistance, but with inductances increasing with depth.

Under running conditions, the slip is quite small and the frequency of the rotor currents is only 1-2 Hz. As a result, the leakage reactance is neglected, and the current distribution is essentially uniform throughout each rotor bar. The effective resistance of the rotor is that of all layers in parallel. This low resistance makes for low slip and high efficiency at full load. As the motor is overloaded, however, slip and frequency increase. Thus, the value of rotor leakage reactance in the circuit model becomes smaller and the value of rotor resistance becomes greater.

Another important need is to include the leakage path saturation effect into a reduced model. The main saturation leakage path is at the tooth tips over the closed or nearly closed slots [46]. However, large machines usually have large slots openings, and hence the leakage path saturation effect is not significant for large machines [3].

Numerous approaches regarding the modelling of deep-bar effect have been presented in literature; therefore, all these models are complicated and require complex mathematics [3], [41], [42].

In [3] is presented a simplified approach for the modelling of deep bar effect, which gives accurate results. The equivalent rotor winding resistance and leakage reactance are modelled as follows:

![Deep-bar effect diagram](image-url)
\[ R'_{r_{\text{DC}}} = R'_{r_{\text{DC}}} + \left( \frac{R'_{r_{\text{50Hz}}} - R'_{r_{\text{DC}}}}{s} \right) \cdot s \]
\[ X'_{\text{sr_{DC}}} = X'_{\text{sr_{DC}}} + \left( \frac{X'_{\text{sr_{50Hz}}} - X'_{\text{sr_{DC}}}}{s} \right) \cdot s \]

where:
- \( R'_{r_{\text{DC}}} \) is the equivalent rotor winding resistance at zero frequency (DC value or at synchronous speed);
- \( R'_{r_{\text{50Hz}}} \) is the equivalent rotor winding resistance at 50Hz (at standstill the slip \( s=1 \));
- \( X'_{\text{sr_{DC}}} \) is the equivalent stator leakage reactance at zero frequency (DC value or at synchronous speed);
- \( X'_{\text{sr_{50Hz}}} \) is the equivalent stator leakage reactance at 50Hz (at standstill the slip \( s=1 \)).

Therefore, this approach assumes a linear dependency of rotor resistance and rotor leakage reactance against the entire range of slip. Moreover, it requires extra parameters determined from tests at synchronous speed.

Usually, the parameters from data sheets are given for rated operating point, near the synchronous speed and the method cannot be applied.

Based only on the parameters from datasheet, the equivalent rotor winding resistance and leakage inductance at any operating point can be modelled as:

\[
\begin{align*}
R'_{r_{\text{dble}}} & = \begin{cases}
R', & |s| \leq s_m \\
R' \left( K_r - 1 \right)|s| + \frac{1 - K_r s_m}{1 - s_m}, & s > s_m
\end{cases} \\
X'_{\text{sr_{dble}}} & = \begin{cases}
X', & |s| \leq s_m \\
X' \left( K_x - 1 \right)|s| + \frac{1 - K_x s_m}{1 - s_m}, & s > s_m
\end{cases}
\end{align*}
\]

where:
- \( R' \) and \( X' \) are the values the from data-sheet given for rated operating point
- \( K_r \) is the resistance deep-bar factor;
- \( K_x \) is the leakage reactance deep-bar factor;
- \( s_m = R' \sqrt{\frac{R_r^2 + X_s^2}{X_r^2}} \) is the breakdown slip in motoring mode.

This approach assumes that the parameters of induction machine are constant for the values of the slip less or equal with the breakdown slip.

Resistance and leakage reactance deep-bar factors can be determined based on rated voltage and current, starting current and phase angle (from short-circuit test at rated current). If the phase angle is not available, a value around 80° can be used as a good approximation in the first step of iteration. Then it can be corrected so that the calculated starting torque will fit the value from datasheet or test reports.
The deep-bar factors for rotor resistance and rotor leakage reactance are calculated as:

\[ K_r = \frac{1}{R_r} \left( \frac{U_{ph \_rated}}{\beta I_{ph \_rated}} \sqrt{\frac{1}{1 + \tan^2 \phi_{sc}}} - R_s \right) \]

\[ K_x = \frac{1}{X_{sc}} \left[ \left( R_s + K_r R_r \right) \cdot \tan \phi_{sc} - X_{cs} \right] \]  

\[ (4.2.48) \]

where:

- \( U_{ph \_rated} \) is the rated voltage per phase;
- \( I_{ph \_rated} \) is the rated current per phase;
- \( \beta \) is the relative starting current (from data sheet);
- \( R_s \) is the stator resistance per-phase (from data sheet);
- \( X_{cs} \) is the stator leakage reactance per-phase (from data sheet);
- \( R_r \) is the rotor resistance per-phase for the rated operating point (from data sheet);
- \( X_{or} \) is the rotor leakage reactance per-phase for the rated operating point (from data sheet);
- \( \phi_{sc} \) is the phase angle at standstill \( (s=1) \): from short-circuit test at rated current or around 80° for large induction machines.

Fig. 4.22a shows the rotor resistance and the rotor leakage reactance against slip based on (4.2.47) and (4.2.48), while Fig. 4.22b shows the electromagnetic torque with and without deep-bar effect.

![Fig. 4.22](image)

Fig. 4.22. Deep-bar effect for a 2 MW squirrel-cage induction machine as a function of slip: a) rotor resistance rotor leakage reactance and b) electromagnetic torque.

The same approach can be used for doubly fed induction machine since the manufacturers performs the standard tests as for the squirrel-cage induction machines.

The Simulink model of induction machine, which implements the variation of the rotor parameters with the slip, is shown in Fig. 4.23.
The mask interface for this model is shown in Fig. 4.24.

In order to include the deep-bar effect some additional parameters have been added in the mask interface: rated voltage and current, base frequency, number of pole pairs, starting current, phase angle at standstill. All these parameters are usually available in standard data-sheets.
4.2.1.6 Modelling saturation

The flux equations highlight the terms, which are dependent on magnetic saturation and therefore introduce non-linearity. It will be assumed that the saturation of the leakage fluxes into a large induction machine can be neglected [33], [46] thus, the stator and rotor leakage inductances are constant.

The magnetizing inductance is a function of magnetization current as shown in Fig. 4.25.

\[ \frac{d}{dt}(\Psi_m) = \frac{d}{dt}(L_m \cdot i_m) = \frac{d}{dt}\left(\frac{L_m \cdot i_m}{\frac{d}{dt}i_m}\right) \frac{di_m}{dt} \]  \hspace{1cm} (4.2.49)

The magnetization current is independent on the reference frame. Based on magnetization curve shown in Fig. 4.25 two inductances can be defined as follows:

\[ L_m = \frac{\Psi_m}{i_m} = \tan \alpha_1 = f_1(i_m) \]  \hspace{1cm} (4.2.50)

\[ L_d = \frac{d}{dt}(L_m \cdot i_m) = \frac{d\Psi_m}{di_m} = \tan \alpha_2 = f_2(i_m) \]  \hspace{1cm} (4.2.51)

where: \( L_m \) is the magnetizing inductances and \( L_d \) is the dynamic inductance.

Fig. 4.26 shows some typical dependencies of these two inductances with the magnetizing current.
Based on the no-load curve for an induction machine these two inductances can be determined as a function of the no-load current. Then, using a mathematical approximation e.g. a function or a look-up table, the desired values for these inductances for a given current can be found.

Taking into account the above mentioned, the phasor voltage equations of the induction machine in the general reference frame including the saturation of the main path can be written as follow:

\[
\begin{align*}
\vec{v}_s &= R_s \vec{i}_s + L_{os} \frac{d}{dt}(\vec{i}_s) + L_d \frac{d\vec{i}_m}{dt} + j \omega_g (L_{os} \vec{i}_s + L_m \vec{i}_m) \\
\vec{v}_r &= R_r \vec{i}_r + L_{or} \frac{d}{dt}(\vec{i}_r) + L_d \frac{d\vec{i}_m}{dt} + j (\omega_g - \omega_r) (L_{or} \vec{i}_r + L_m \vec{i}_m) \\
\vec{i}_m &= \vec{i}_s + \vec{i}_r \\
L_m &= f_1(\vec{i}_m) \\
L_d &= f_2(\vec{i}_m)
\end{align*}
\]

(4.2.52)

### 4.2.1.7 Modelling iron losses

Considerable attention has been paid during the last years in modelling of iron losses into an induction machine. A simple search in IEEE database will produce around 120 results. All these papers deals with the iron loss modelling for an inverter-fed induction motor and take into account the hysteresis losses as well as the eddy-current losses in a wide range of speed and frequency. In wind turbine applications usually a large squirrel-cage induction generator is directly connected to grid and the rotor speed is almost fixed. When a doubly fed machine is used, again the stator windings are directly connected to the grid and the variable speed operation is achieved using a power converter in the rotor side. Usually, the speed varies in a range of ± 30% from the synchronous speed. In data sheets the no-load power, current and power factor are provided for the machine rated voltage as well as the electrical parameters for the rated operating point. So, it is convenient to use only this information in the case of a large induction machine, both squirrel-cage and wound rotor.

A classical approach in iron losses modelling for an induction machine involves a resistance inserted in parallel with the magnetizing reactance in the equivalent per-phase diagram, as shown in Fig. 4.27.
One major drawback of this approach is that the magnetizing current is divided into two components: an active component and a reactive component. So, it is difficult to use this equivalent diagram for a dynamic model as well as for a steady state analysis. A modified equivalent scheme, called $\Gamma$, has been developed. Unfortunately, the currents in this equivalent scheme are multiplied with some factors and the dynamic modelling become difficult.

A simple approach assumes the iron losses resistance in series with the magnetizing reactance as shown in Fig. 4.28.

Where the parameters of the magnetizing branch can be determined based on parameters from Fig. 4.27 as follows:

$$R_o = \frac{R_{ix}}{R_{ix}^2 + X_{im}^2} - \text{equivalent series resistance;}$$

$$X_o = \frac{R_{ix}}{R_{ix}^2 + X_{im}^2} - \text{equivalent series reactance.}$$

Since the machine manufacturers give the electrical parameters for both schemes presented in Fig. 4.27 and Fig. 4.28 in the data sheet, or at least for the first one it is easier to use this approach. The electrical parameters for the parallel mode can be transformed through some simple mathematical calculations to the series mode equivalent circuit.

Recalling (4.2.1) the iron losses can be introduced into the dynamic equations written in the $abc$ reference frame as follows:

$$[V] = [R][i] + [R_o][i] + \frac{d}{dt} [\Psi]$$

where the iron losses resistance matrix is defined in a similar manner as (4.2.6) by:

$$[R_o] = R_o \cdot \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = R_o \cdot [T_o]$$

In this condition the inductance matrix (4.2.10) will be rewritten as:
\[
[L] = \begin{bmatrix}
L_s & 0 & 0 & M_o f_1 & M_o f_2 & M_o f_3 \\
0 & L_s & 0 & M_o f_3 & M_o f_1 & M_o f_2 \\
0 & 0 & L_s & M_o f_2 & M_o f_3 & M_o f_1 \\
M_o f_1 & M_o f_3 & M_o f_2 & L_r & 0 & 0 \\
M_o f_2 & M_o f_1 & M_o f_3 & 0 & L_r & 0 \\
M_o f_3 & M_o f_2 & M_o f_1 & 0 & 0 & L_r
\end{bmatrix}
\]  \hspace{1cm} (4.2.55)

where:
- \( L_s = L_{cs} + \frac{3}{2} M_o \) - stator self-inductance;
- \( L_r = L_{cr} + \frac{3}{2} M_o \) - rotor self-inductance;
- \( M_o \) - magnetizing inductance for the series circuit.

Finally the state space equations with fluxes as state variables are:
\[
\frac{d}{dt}[\Psi] = -([R] + [R_o])[L]^{-1}[\Psi] + [V] \hspace{1cm} (4.2.56)
\]
or using currents as state variables:
\[
\frac{d[I]}{dt} = [L]^{-1} \cdot \left\{ -([R] + [R_o]) - \omega_c \cdot \frac{d[L]}{d\theta_c} \right\} \cdot [I] + [L]^{-1} \cdot [V] \hspace{1cm} (4.2.57)
\]

A similar approach can be used to introduce the iron loss resistance into the dynamic equations written in arbitrary reference frame.

### 4.2.2 Synchronous Machine Model

In the following the dynamic equations for salient-pole synchronous machines are presented.

#### 4.2.2.1 Dynamic Equations of Synchronous Machine

The dynamic equations of synchronous machine are derived for a salient pole machine with damping bars, which is the most complicated model for this type of machine [7], [9], [21], [33], [34], [46], [75]. This model can be also used for a non-salient pole machine assuming identical values for inductances in \( d \)- and \( q \)-axis.

**Untransformed model**

In synchronous machines with salient poles the damping bars can be represented by a direct- and a quadrature-axis damping winding, which are short-circuited windings. Assume that on the stator of the machine there is a symmetrical three-phase sinusoidal distributed stator winding and on the rotor there are the field winding and direct- and quadrature-axis damper winding. Schematic of the machine is shown in Fig. 4.29.
The phasor form of the stator voltage equation in the stationary reference frame is:

\[
\tilde{u}_{sA,B,C} = R_s \cdot \tilde{i}_{sA,B,C} + \frac{d\psi_{sA,B,C}}{dt}
\]  

(4.2.58)

where \( R_s \) is the resistance of the stator winding.

The voltage equations of the field winding and the damper windings in the reference frame fixed to the rotor are as follows:

\[
\begin{align*}
    u_f &= R_f \cdot i_f + \frac{d\psi_f}{dt} \\
    u_D &= R_D \cdot i_D + \frac{d\psi_D}{dt} \\
    u_Q &= R_Q \cdot i_Q + \frac{d\psi_Q}{dt}
\end{align*}
\]  

(4.2.59)

**Transformed model with quadrature-phase stator windings**

Replacing the three-phase stator winding by an equivalent quadrature-phase stator winding, the number of differential equations can be reduced. Schematic of the corresponding model is shown in Fig. 4.30.

The stator voltage equations are:
The voltage equations of the field winding and the damper windings in the reference frame fixed to the rotor are the same.

**Commutator model**

It can be shown that when the stator quantities of the salient pole machine are expressed in the d-q-o reference frame fixed to the rotor, the components of the stator flux linkages expressed in the rotor reference frame will not contain the rotor angle. This corresponds to the commutator model of the salient-pole machine. This approach is useful for studying the unsymmetric behaviour of the synchronous machine. It follows that in the rotor reference frame, under linear conditions, the salient pole machine can be described by a system of voltage equations with constant coefficients and, in general, the only changing quantity is the rotor speed in these equations.

Thus, the stator equations in phasor form in the rotor reference frame are given by:

\[
\vec{u}_s = R_s \vec{i}_s + \frac{d\vec{\Psi}_s}{dt} + j\omega_r \vec{\Psi}_s
\]  
\[
(4.2.61)
\]

where \( \omega_r \) is the rotor speed. The stator flux linkages in the d-q-o rotor reference frame can be expressed in terms of the currents of the machine as:

\[
\psi_{sd} = \left( L_{sd} + M_0 + \frac{3}{2} L_2 \right) i_{sd} + \sqrt{\frac{3}{2}} M_f i_f + \sqrt{\frac{3}{2}} M_D i_D
\]

\[
\psi_{sq} = \left( L_{sq} + M_0 - \frac{3}{2} L_2 \right) i_{sq} + \sqrt{\frac{3}{2}} M_Q i_Q
\]

\[
\psi_{so} = \left( L_{so} - 2 M_0 \right) i_{so}
\]  
\[
(4.2.62)
\]

We note:

\[
L_d = \left( L_{sd} + M_0 + \frac{3}{2} L_2 \right) - \text{direct-axis synchronous inductance}
\]

\[
L_q = \left( L_{sq} + M_0 - \frac{3}{2} L_2 \right) - \text{quadrature-axis synchronous inductance};
\]

\[
L_o = L_{so} - 2 M_0 - \text{homopolar inductance};
\]

\[
M_f = \sqrt{\frac{3}{2}} M_f - \text{the mutual inductance between the field winding and the stator winding};
\]
\[ M_D' = \frac{3}{\sqrt{2}} M_D \] - the mutual inductance between the stator winding and the direct-axis damper winding.

\[ M_Q' = \frac{3}{\sqrt{2}} M_Q \] - the mutual inductance between the stator winding and the quadrature-axis damper winding.

The synchronous inductances can be expressed as follows:

\[ L_d = L_{cs} + L_{md} \]
\[ L_q = L_{cs} + L_{mq} \]

where

\[ L_{cs} \] - the stator leakage inductance;

\[ L_{md} = M_0 + \frac{3}{2} L_2 \] - the magnetizing inductance in direct-axis;

\[ L_{mq} = M_0 - \frac{3}{2} L_2 \] - the magnetizing inductances in quadrature-axis.

Splitting (4.2.61) into real and imaginary part yields the following dqo equations for the stator are obtained:

\[ u_{sd} = R_s \cdot i_{sd} + \frac{d\psi_{sd}}{dt} - \omega_f \cdot \psi_{sq} \]
\[ u_{sq} = R_s \cdot i_{sq} + \frac{d\psi_{sq}}{dt} - \omega_f \cdot \psi_{sd} \] (4.2.63)
\[ u_{so} = R_s \cdot i_{so} + \frac{d\psi_{so}}{dt} \]

where

\[ \psi_{sd} = L_{ds} i_{sd} + M'_f i_f + M'_D i_D \]
\[ \psi_{sq} = L_{qs} i_{sq} + M'_Q i_Q \]
\[ \psi_{so} = L_{os} i_{so} \]

The voltage equations of the field winding and the direct- and quadrature-axis damper windings can be obtained in the reference frame fixed to the rotor as:

\[ u_f = R_f \cdot i_f + \frac{d\psi_f}{dt} \]
\[ u_D = R_D \cdot i_D + \frac{d\psi_D}{dt} \] (4.2.64)
\[ u_Q = R_Q \cdot i_Q + \frac{d\psi_Q}{dt} \]
where $R_f$, $R_D$ and $R_Q$ are the resistances of the field winding and direct- and quadrature-axis damper windings respectively, and the flux linkages of the field winding and the damper windings are obtained as:

$$
\psi_f = L_f i_f + M_{f D} i_D + M_{f Q} i_{sd}
$$

$$
\psi_D = L_D i_D + M_{f D} i_f + M_{D Q} i_{sd}
$$

$$
\psi_Q = L_Q i_Q + M_{Q Q} i_{sq}
$$

where

- $L_f$ - self-inductance of the field winding;
- $L_D$, $L_Q$ - self-inductances of the damper winding in the direct- and quadrature-axis;
- $M_{f D}$ - mutual inductance between field winding and direct damper winding

Thus the matrix form of the voltage equations of the commutator model are obtained as follows:

$$
\begin{bmatrix}
    u_{sd} \\
    u_{sq} \\
    u_{so} \\
    u_f \\
    u_D \\
    u_Q
\end{bmatrix} =
\begin{bmatrix}
    R_s + pL_d & -\omega_r L_q & 0 & pM'_{f} & pM'_{D} & \omega_r M'_{Q} \\
    \omega_r L_d & R_s + pL_q & 0 & \omega_r M'_{f} & \omega_r M'_{D} & pM'_{Q} \\
    0 & 0 & R_s + pL_o & 0 & 0 & 0 \\
    pM_{f D} & 0 & 0 & R_f + pL_f & pM'_{f} & 0 \\
    pM'_{D} & 0 & 0 & pM_{f D} & R_D + pL_D & 0 \\
    0 & pM'_{Q} & 0 & 0 & 0 & R_Q + pL_Q
\end{bmatrix}
\begin{bmatrix}
    i_{sd} \\
    i_{sq} \\
    i_{so} \\
    i_f \\
    i_D \\
    i_Q
\end{bmatrix}
$$

(4.2.66)

In case of symmetrical and balanced operation of the synchronous machine the homo-polar component of the stator current is zero, so the homo-polar component of the stator flux is zero.

**Electromagnetic torque**

The general expression of the electromagnetic torque is:

$$
T_e = \frac{3}{2} p (\Psi_s \times \bar{i}_s)
$$

(4.2.67)

Taking into account the expressions of the stator flux and stator current in the rotor reference frame the electromagnetic torque can be written as follows:

$$
T_e = \frac{3}{2} p (\psi_{sd} i_{sq} - \psi_{sq} i_{sd}) = \frac{3}{2} p (L_d - L_q) i_{sd} i_{sq} - M_{Q Q} i_{sq} i_{sd} + (M_{f f}' + M_{f D} i_D) i_{sq}
$$

(4.2.68)

**4.2.2.2 Simulink model**

The dynamic equations (4.2.66) and (4.2.68) are implemented in Matlab/Simulink, as shown in Fig. 4.31.
In Fig. 4.32 is shown the mask interface for this model in Simulink.

Since the model is written for a salient pole synchronous machine, the mask parameters include different values for the rotor resistances and leakage inductances in $d$ and $q$-axis as well as for the magnetizing inductances. An extra input are the parameters for the field winding.

### 4.2.3 Permanent-magnet synchronous machine

The analysis of permanent magnet (PM) synchronous machines can be made using an quadrature equivalent circuit where the damping windings are replaced with two equivalent windings $D$, $Q$ in direct and quadrature axis respectively and the permanent magnet is replaced with an equivalent superconductor winding placed in the direct-axis (Fig. 4.33). The current through the equivalent winding of the permanent magnet ($I_F$) is constant in all mode of operation.
The voltage equations of the PM synchronous machines in $dqo$ rotor reference frame can be derivated from the voltage equation of synchronous machine (2.26) taking into account the above assertions.

The voltage equations for PM synchronous machine can be written as:

\[
\begin{align*}
\frac{du_s}{dt} &= R_j_i_s + \frac{d\psi_s}{dt} - \omega_x \psi_q \\
\frac{du_q}{dt} &= R_j_i_q + \frac{d\psi_q}{dt} + \omega_x \psi_s \\
\frac{du_o}{dt} &= R_j_i_o + \frac{d\psi_o}{dt} \\
\frac{du_d}{dt} &= R_j_i_d + \frac{d\psi_d}{dt} \\
\frac{du_{sq}}{dt} &= R_j_i_{sq} + \frac{d\psi_{sq}}{dt} \\
\frac{du_{so}}{dt} &= R_j_i_{so} + \frac{d\psi_{so}}{dt} \\
\frac{du_{sd}}{dt} &= R_j_i_{sd} + \frac{d\psi_{sd}}{dt}
\end{align*}
\]  

(4.2.69)

The flux linkage equations can be written as:

\[
\begin{align*}
\psi_{sd} &= L_d i_{sd} + M_{fq} i_f + M_{dq} i_D \\
\psi_{sq} &= L_q i_{sq} + M_{qf} i_q \\
\psi_{so} &= L_0 i_{so} \\
\psi_f &= L_f i_f = c_t. \\
\psi_D &= L_D i_D + M_{Dq} i_q + M_{Df} i_s \\
\psi_Q &= L_Q i_Q + M_{Qf} i_s
\end{align*}
\]  

(4.2.70)
The matrix form of the voltage equations for PM synchronous machines in dqo rotor reference frame is:

\[
\begin{bmatrix}
    u_{sd} \\
    u_{sq} \\
    u_{so} \\
    0 \\
    0 = u_D \\
    0 = u_Q
\end{bmatrix} =
\begin{bmatrix}
    R_s + pL_d & -\omega_r L_q & 0 & 0 & pM_D' & -\omega_r M_Q' \\
    \omega_r L_d & R_s + pL_q & 0 & \omega_r M_f' & \omega_r M_D' & pM_Q' \\
    0 & 0 & R_s + pL_d & 0 & 0 & 0 \\
    0 & 0 & 0 & R_d + pL_D & 0 & 0
\end{bmatrix}
\begin{bmatrix}
    i_{sd} \\
    i_{sq} \\
    i_{so} \\
    i_D \\
    i_Q
\end{bmatrix}
\]

The term \( pM_f' \) corresponds to the e.m.f. induced by the rotating magnet in the equivalent quadrature-axis stator winding.

The general expression of the electromagnetic torque is given by:

\[
T_e = \frac{3}{2} P (\psi_s \times \mathbf{i}_s) = \frac{3}{2} P (\psi_{sq} i_{sq} - \psi_{sd} i_{sd})
\]

The Simulink implementation of (4.2.71) (4.2.72) is shown in Fig. 4.35.

Fig. 4.35. Simulink model for permanent magnet synchronous machine.

The model take into account the iron losses as well as the parameters variation with the operating temperature.

The mask interface for this model is presented in Fig. 4.36. Notice that the operating temperature is an input parameter in the model and there is also the possibility to take the iron losses into account.
4.3 Power converter models

In this paragraph some aspects regarding the modelling of the power converter topologies for wind turbine applications will be presented.

The focus is on the modelling of different soft-starter-fed induction machine topologies as well as on the voltage source converters modelling using the switching function concept. The influence of the load connection is taken into account in both cases.

4.3.1 AC Controllers (Soft-starters)

Many authors deal with the theory of operation for three-phase AC-controllers (soft-starters), [67], [69], and [73]. However, only few of them present in details the theory of operation with a 3-phase resistive-inductive load [67] and [73]. Especially the influence of the load connection type and the steady-state analysis are not presented in detail in the literature.

In [70] a comparison for full and half-wave controlled load both for star and delta connection is carried out, unfortunately, the branch delta connection is not treated. An analysis of variable-voltage thyristor controlled induction motors in terms of specific operation modes, harmonic content and performances for a star connected machine is presented in [26]. In [74] is presented a hybrid abc-dqo model for the induction machine, unfortunately this model has a complicated mathematical model and it can be used only for a star connection of the stator windings. A generalized approach in modelling of the power converters-fed induction machine, which involves a time-domain static network, is presented in [22]. In order to eliminate the computation errors this model uses some “dummy” shunt resistances. However, the proposed method can only be used for a star or delta-connected machine. Moreover these approaches do not take into account the deep-bar effect, which is present into a large induction machine.
Therefore in analysis of a soft-starter-fed induction machine is very convenient to use the abc model, since it is based on per-phase quantities.

At present many wind turbines, up to 2.3 MW, are based on the “Danish concept” in which a squirrel-cage induction generator is directly connected to the grid as shown in Fig. 4.37.

![Fig. 4.37. Block diagram of directly grid-connected wind turbine.](image)

The scheme comprises the wind turbine rotor, linked via a gearbox to a generator, which through an electrical interface is connected to the grid. A control system is necessary to assure a proper operation of the wind turbine under all conditions. The electrical interface consists of: a soft-starter, a capacitor bank and a transformer, which makes the connection with the medium voltage grid. The capacitor bank is used to control the power factor of the generator output.

Soft-starters are used only during the start-up sequence of the generator in order to limit the inrush currents and the starting torque transients in the drive train.

There are many configurations of soft-starters, which feed an induction machine. However there are three topologies interesting for wind turbine applications, as presented in Fig. 4.38.

![Fig. 4.38. Possible configurations of soft-starter-fed induction machine in wind turbine applications: a) star connection, b) delta connection and c) branch-delta connection.](image)

The star and delta configurations have basically the same layout for the semiconductors (SCRs), the difference consists in the machine winding connection. There are two anti-parallel thyristors on each phase, each one conducting on the positive cycle of the applied voltage. For a star connection the applied per phase voltages depend on the on-state of the SCRs on each phase. So, the soft-starter can operate only when two or three SCRs are conducting in the same moment. The similar considerations can be done for a delta connection.

However a soft-starter with a branch delta connection, will operate only with one SCR conducting at a given moment.
In wind turbine applications mainly the delta connection for the induction machine are used because the current rating of the stator windings can be reduced, and the third harmonic in the line currents is eliminated in this case.

Depending on the firing angle $\alpha$, three different modes of operation of the soft-starter can be distinguished when a star or delta connected resistive load is used [67]:

- Mode 1 - $0 \leq \alpha < 60^\circ$ - two or three SCRs are conducting (in either direction);
- Mode 2 - $60^\circ \leq \alpha < 90^\circ$ - two SCRs are conducting;
- Mode 3 - $90^\circ \leq \alpha < 150^\circ$ - none or two SCRs are conducting.

where $\alpha$ is the firing angle for the soft-starter.

When a resistive-inductive load is used the analysis of the controller is difficult, since the operation modes depend on the extinction angle $\xi$ and the limit angle $\alpha_{\text{lim}}$, both dependent on the phase angle $\phi$.

Mode 2 of operation, characterized by rapid changes of the output current, is not possible due to the load inductance. The ranges of the two remaining operation modes are:

- Mode 1: $\varphi \leq \alpha < \alpha_{\text{lim}}$ - two or three SCRs are conducting;
- Mode 3: $\alpha_{\text{lim}} \leq \alpha < 150^\circ$ - none or two SCRs are conducting.

The limit angle can be determined numerically from:

$$\sin\left(\alpha_{\text{lim}} - \varphi - \frac{4\pi}{3}\right) = \frac{2e^{\frac{\pi}{3\tan\varphi}} - 1}{2 - e^{\frac{-\pi}{3\tan\varphi}}}$$

(4.3.1)

Switching functions $S_{wA}$, $S_{wB}$, and $S_{wC}$ in two levels can be introduced in modelling of the SCRs and defined as equal to one when a given thyristor is conducting and equal to zero otherwise as shown in Fig. 4.39.

These switching functions can be determined based on phase voltage and phase (or line) current for each particular topology.

**4.3.1.1 Star connected 3-phase load**

A soft-starter-fed 3-phase resistive-inductive load in star connection is shown in Fig. 4.40.
Analyzing this topology the applied per-phase voltages at the machine terminals can be written as:

\[
\begin{bmatrix}
 v_{ux} \\
 v_{vy} \\
 v_{wz}
\end{bmatrix} = \frac{1}{2} \begin{bmatrix}
 S_{wA} & -S_{wB} & -S_{wC} \\
 -S_{wA} & S_{wi} & -S_{wC} \\
 -S_{wA} & -S_{wB} & S_{wC}
\end{bmatrix} \begin{bmatrix}
 v_{AN} \\
 v_{BN} \\
 v_{CN}
\end{bmatrix}
\]  

(4.3.2)

where: \([v_{ux} \ v_{vy} \ v_{wz}]^T\) are the phase voltages of the machine; \([v_{AN} \ v_{BN} \ v_{CN}]^T\) are the phase-to-ground supply voltages and \(S_{wA}, S_{wB}, S_{wC}\) are the switching functions for each phase.

Based on the phase-to-ground voltages and the phase (line) currents the switching functions are computed.

The considered load has a power factor of 0.85. Using (4.3.1) a limit angle around 95° will be the result. In order to highlight the two possible operation modes for the soft-starters two values for the firing angle have been used. A value of 70° for the firing angle will correspond to Mode 1 and 110° to Mode 3 respectively.

The per-phase voltage and the phase current for the considered load are shown in Fig. 4.41 for different firing angles.

From Fig. 4.41 it can be observed that when the firing angle \(\alpha\) is greater than the limit angle \(\alpha_{\text{lim}}\) the phase currents occurs in discontinuous, non-sinusoidal, alternating pulses. Therefore, a high harmonic content for the phase currents will be the result.
4.3.1.2 Delta connected 3-phase load

Fig. 4.42 shows a three-phase resistive-inductive load in delta connection.

It can easily be demonstrated that the voltages applied to a delta-connected load are given by:

\[
\begin{bmatrix}
V_{UX} \\
V_{ZY} \\
V_{WZ}
\end{bmatrix} =
\begin{bmatrix}
S_{wA} S_{wB} & -0.5 S_{wA} & -0.5 S_{wB} \\
-0.5 S_{wC} & S_{wB} & S_{wC} \\
-0.5 S_{wC} & -0.5 S_{wA} & S_{wA}
\end{bmatrix}
\begin{bmatrix}
V_{AB} \\
V_{BC} \\
V_{CA}
\end{bmatrix}
\]

(4.3.3)

where: \([V_{AB}, V_{BC}, V_{CA}]^T\) are the line-to-line supply voltages (or phase voltages for the load).

For a delta connection the switching functions are determined based on phase-to-ground voltages and line currents.

Considering the same load as in the precedent case the applied per-phase voltage, the phase and the line currents for different firing angles are shown in Fig. 4.43.

Again, it can be observed that when the firing angle \(\alpha\) is greater than the limit angle \(\alpha_{lim}\) the line currents occurs in discontinuous, non-sinusoidal, alternating pulses. Therefore, for this topology a high harmonic content for both line and phase currents will result especially in Mode 3. As a result the electromagnetic torque will have a high harmonic content.
4.3.1.3 Branch-delta connected load

A soft-starter-fed 3-phase resistive-inductive load in branch-delta connection is presented in Fig. 4.44.

The output voltages of the soft-starter in this case are:

\[
\begin{bmatrix}
  v_{UX} \\
  v_{YV} \\
  v_{WZ}
\end{bmatrix} =
\begin{bmatrix}
  S_{wA} & 0 & 0 \\
  0 & S_{wB} & 0 \\
  0 & 0 & S_{wC}
\end{bmatrix}
\begin{bmatrix}
  v_{AB} \\
  v_{BC} \\
  v_{CA}
\end{bmatrix}
\] (4.3.4)

For a branch-delta connection the switching functions are determined based on the line voltages and line currents.

Again, a 3-phase resistive-inductive load as in precedent cases has been considered. The per-phase voltage, the phase and the line current are shown in Fig. 4.45 for the considered firing angles.

Using a branch-delta connection the line current waveform is approximately the same for the entire range of possible firing angle. Therefore, the harmonic content in the line / phase currents as well as in the electromagnetic torque is not so high comparing with the other two topologies.
4.3.1.4 Simulink implementation of the soft-starter

The soft-starter model including the connection type as well as the algorithm for generation of the switching function has been implemented in Simulink as shown in Fig. 4.46.

The mask interface for this model is shown in Fig. 4.47. The input parameters for this model are the fundamental frequency of the grid voltages and the initial states for the switches. The selection of the connection type is also an option in this mask.

4.3.1.5 RMS model for soft-starter

In steady-state analysis of the soft-starter-fed induction machine or when reduced order models of the machine are used, an RMS model for the AC controller should be used. The expression of the soft-starter output voltage is a function of firing angle, phase angle and so-called extinction angle.

Since, a SCR will not permit the flow of reverse current, the conduction of it will end at a point \( \xi \), which is called extinction angle or cut-off angle. The extinction angle can be obtained solving the following transcendental equation [73]:

\[
\sin(\xi - \phi) - \sin(\alpha - \phi) e^{-\alpha \phi (\xi - \alpha)} = 0 \tag{4.3.5}
\]

where: \( \alpha \) is the firing angle and \( \phi \) is the phase angle (power factor).
Only an iterative solution of \((4.3.5)\) is possible. This yields the set of characteristics shown in Fig. 4.48.

![Extinction angle](image)

Fig. 4.48. Extinction angle versus firing angle for a series R-L load with different power factors.

When \(\alpha = \phi\), which represents sinusoidal operation, \((4.3.5)\) reduces to \(\sin(\xi - \phi) = 0\), which gives \(\xi = \pi + \phi\). The values \(\xi\) for sinusoidal operation are seen to lie on the dashed linear characteristic of Fig. 4.48. For a purely inductive load, therefore, the variation of \(\xi\) with \(\alpha\) is linear, as shown in the \(\phi = 90^\circ\) characteristic from Fig. 4.48.

Considering a single R-L load as shown in Fig. 4.49

![Single-phase controller](image)

Fig. 4.49. Single-phase controller with a resistive inductive load.

the applied voltage at the load terminals is given by \([73]\):

\[
V_{\text{UX}} = V_{\text{AB}} \sqrt{\frac{1}{2\pi} \left( (\xi - \alpha) + 0.5 \sin 2\alpha - 0.5 \sin 2\xi \right)} \tag{4.3.6}
\]

Based on \((4.3.6)\) the variation of the fundamental voltage with firing angle is shown in Fig. 4.50 for several fixed values of phase-angle.

Equation \((4.3.6)\) is valid only for a branch-delta connection. Based on the input voltage and the phase current in complex form and a look-up table for the extinction angle the output voltage per-phase can be obtained.

Unfortunately, for the star or the delta connection due to the complexity of the conducting mechanism it is very difficult to establish an analytical expression for the RMS AC-controller output voltage.

The Simulink implementation of \((4.3.6)\) is based on a look-up table for the extinction angle as shown in Fig. 4.51.
4.3.2 Voltage Source Converters

Currently, in wind turbine applications the back-to-back voltage source converter (VSC) is mainly used. A block diagram of this power converter is shown in Fig. 4.52.

This topology comprises a double conversion from AC to DC and then from DC to AC. Both converters can operate in rectifier or inverter mode and therefore a bi-directional power flow can be achieved.

A voltage source converter can be implemented in several ways: six-step, pulse amplitude modulated (PAM) or pulse width modulated (PWM). Moreover, the implementation of a
PWM VSC may be realized by three methods: harmonic elimination, “sinusoidal” PWM or space vector strategy (SV-PWM).

Independent on the implementation method the converter can be seen as a black box with some input-output characteristics as a function of the control strategy. Again, the switching function concept can be used in modelling of these topologies.

So, in the following paragraphs the input-output characteristics as a function of the switching function for VSCs, both inverter and rectifier mode of operation will be presented.

The influence of the windings connection for generator will be also taken into account.

### 4.3.2.1 Star connected generator

When a Voltage Source Converter is used in conjunction with a star connected machine (squirrel-cage or wound rotor) the equivalent diagram of the circuit is like in Fig. 4.53.

![Induction Generator](image)

The applied voltages at the machine terminals as a function of the DC-link voltage and the switching functions are [48]:

\[
\begin{bmatrix}
v_{UX} \\
v_{VY} \\
v_{WZ}
\end{bmatrix} = \frac{1}{3} V_{DC} \begin{bmatrix}
2 & -1 & -1 \\
-1 & 2 & -1 \\
-1 & -1 & 2
\end{bmatrix} \begin{bmatrix}
S_{wA} \\
S_{wB} \\
S_{wC}
\end{bmatrix}
\]

The DC-link current can be expressed as a function of the input currents and the switching functions by:

\[
i_{DC} = \begin{bmatrix} S_{wA} & S_{wB} & S_{wC} \end{bmatrix} \begin{bmatrix} i_A \\
i_B \\
i_C \end{bmatrix}
\]

The voltages and currents waveforms in this case are shown in Fig. 4.54.
4.3.2.2 **Delta connected generator**

Fig. 4.55 shows a VSC connected to an induction generator with delta-connected windings.

The applied voltages at the machine terminals as a function of the DC-link voltage and the switching functions are:

\[
\begin{bmatrix}
    v_{UX} \\
    v_{YY} \\
    v_{WZ}
\end{bmatrix}
= V_{DC}
\begin{bmatrix}
    1 & -1 & 0 \\
    0 & 1 & -1 \\
    -1 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    S_{wA} \\
    S_{wB} \\
    S_{wC}
\end{bmatrix}
\]  

(4.3.9)

The DC-link current can be expressed as a function of the input currents and the switching functions by:
\[ i_{DC} = \begin{bmatrix} S_{wA} & S_{wB} & S_{wC} \end{bmatrix} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} = \begin{bmatrix} S_{wA} & S_{wB} & S_{wC} \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} i_{UX} \\ i_{VY} \\ i_{WZ} \end{bmatrix} \] (4.3.10)

The voltages and currents waveforms in the case of a VSC with a delta-connected generator are shown in Fig. 4.56.

Fig. 4.56. Typical waveforms for voltages and currents for a Voltage Source Converter with a delta connected generator.

### 4.3.2.3 Simulink implementation of the voltage source converter

The Simulink model for the voltage source power converter has been implemented based on (4.3.7)-(4.3.10) as shown in Fig. 4.57.

Fig. 4.57. Simulink implementation of the voltage source converter including connection type.
4.3.3 DC-link Circuit

Typically, the DC-link circuit is represented as an ideal capacitor as shown in Fig. 4.58.

The DC-link voltage, which is the voltage at the capacitor terminals, is given by:

\[ v_{DC} = \frac{1}{C} \int i_C dt = \frac{1}{C} \int (i_R - i_I) dt \] (4.3.11)

where \( i_R \) is the current of the generator side converter and \( i_I \) is the current of the grid side converter.

In s-plane modelling (using transfer functions) (4.3.11) cannot be used due to numerical problems. Hence, the capacitor bank should be modelled as a series R-C circuit, which is similar with a real component. A capacitor always will exhibit an internal resistance.

So, (4.3.11) can be written in the s-plane as:

\[ V_{DC} = \left( R + \frac{1}{sC} \right) (I_R - I_I) = \frac{sRC + 1}{sC} (I_R - I_I) \] (4.3.12)

In this way the connection between the two considered VSCs could be achieved through the DC-link circuit as shown in Fig. 4.59.

4.4 Modulation strategies

In this paragraph two modulation strategies for power converters, which can be used in wind turbine applications are presented. These algorithms have been implemented in Matlab/Simulink in two versions using the Simulink Blocks and based on a C S-Function. Both algorithms can be used for a switching model of the power converter as well as with an average model of it. In the last case the duty cycles are applied directly to the converter model.
4.4.1 PWM sinusoidal modulator using sine-triangle with third harmonic

The normalized duty cycle functions for a Sine-Triangle with Third Harmonic (STTH) modulation algorithm are:

\[
\begin{align*}
\text{duty}_a &= 0.5 \cdot m \cdot \frac{2}{\sqrt{3}} \left[ \sin(\theta) + \frac{\sin(3\theta)}{6} \right] + 0.5 \\
\text{duty}_b &= 0.5 \cdot m \cdot \frac{2}{\sqrt{3}} \left[ \sin\left(\theta - \frac{2\pi}{3}\right) + \frac{\sin(3\theta)}{6} \right] + 0.5 \\
\text{duty}_c &= 0.5 \cdot m \cdot \frac{2}{\sqrt{3}} \left[ \sin\left(\theta - \frac{4\pi}{3}\right) + \frac{\sin(3\theta)}{6} \right] + 0.5 
\end{align*}
\] (4.4.1)

where: \( \theta \)- the reference angle is calculated based on the desired output frequency and \( m \) is the modulation index \( m = \sqrt{3} \frac{V_{dc}}{V_{ref}} \).

Equation (4.4.1) can be implemented in Simulink in different ways. However, in order to minimize the number of blocks, which are used for this model, a possible implementation is shown in Fig. 4.60.

After the calculation of the duty cycles for each phase, some calculation are made in order to add the pulse dropping option and to transform these signals into switching signals for the converter legs (0...1). The duty cycles are compared with a triangular carrier signal at the desired switching frequency and then the switching signals are obtained.

Fig. 4.61. Mask interface in Simulink for the STTH modulator.
The input parameters for the mask interface are: carrier frequency, dead-time, minimum pulse width as shown in Fig. 4.61. The model can also take into account the pulse dropping option and the insertion of the third harmonic.

### 4.4.2 Space Vector Modulation

Space Vector Modulation (SVM) strategy simultaneously performs the waveform generation of all three phases within a two-dimensional vectorial reference frame, eliminating the computational redundancy of considering each phase separately. SVM is based on the concept of approximating the reference voltage vector defined as:

\[ V_{ref} = \frac{2}{3} \left( v_a + v_b \cdot e^{j\frac{2\pi}{3}} + v_c \cdot e^{-j\frac{2\pi}{3}} \right) \]  

by one switching-cycle time-averaged combination of two adjacent physically realisable voltage vectors in a three-phase voltage source converter.

![Fig. 4.62. Voltage vectors and space vectors.](image)

The two adjacent voltage vectors are chosen from \( V_1 - V_6 \) according to the six active switching states ([011], [001], [101], [100], [110], [010]), plus two null voltage vectors, i.e. \( V_0 \) and \( V_7 \), corresponding to the passive switching states ([000] and [111]). Switching state [111] means all three upper transistors are switched on.

For high switching frequencies, the reference voltage vector can be considered constant during each switching period. The on-duration of active vectors \( V_1 \) and \( V_2 \) that are adjacent to \( V_{ref} \) are found by solving the vectorial equation (Fig. 4.62):
\[ \begin{align*} 
V_{\text{ref}} \cdot \frac{T_0}{2} &= V_1 \cdot T_1 + V_2 \cdot T_2 \\
T_0 &= \frac{T_{\text{sw}}}{2} - T_1 - T_2 
\end{align*} \]  
\hspace{1cm} (4.4.3)

where \( T_0 \) denoting the time period left from a half switching period is used by the null voltage vectors using a pattern, which ensures that only one leg is switched at every state transition.

Assuming \( V_{\text{ref}} \) is in the first sector, (4.4.3) can be solved in orthogonal d-q axis yielding:

\[ \begin{align*} 
T_1 &= m \cdot \frac{\sqrt{3}}{2} \cdot \frac{T_{\text{sw}}}{2} \left( \cos(\gamma) - \frac{1}{\sqrt{3}} \sin(\gamma) \right) \\
T_2 &= m \cdot \frac{\sqrt{3}}{2} \cdot \sin(\gamma) 
\end{align*} \]

\hspace{1cm} (4.4.4)

where an equivalent modulation index \( m \) was introduced and \( V_{\text{ref}} \) denotes the amplitude of the reference voltage. Therefore, SVM provides accurate control of voltage amplitude, frequency and phase within every switching period being very suitable for field-oriented control.

The Simulink implementation of this modulation strategy is shown in Fig. 4.64. Based on (4.4.3) - (4.4.4) and using the position of the reference voltage phasor \( \gamma \), the duty cycles are calculated for each sector (see Fig. 4.65). Then, the pulse dropping option is included in the model as well as the transformation to switching states and the dead time option.
The mask interface comprises the following input parameters: switching frequency, dead time and minimum pulse width as shown in Fig. 4.66. The pulse dropping option is also a selection in this mask interface.

Alternatively, the model is available in a C S-Function implementation.

4.5 Grid component models

In this paragraph the mathematical models and the Simulink implementation for the subsystems related with the grid connection of the wind turbine are presented. These subsystems are: grid model, cable model, transformer model, circuit breaker model and capacitor bank model.

4.5.1 Grid model

The grid model is based on the Thevenin equivalent representing the transformers and the grid. The equivalent diagram per phase for the grid model is shown in Fig. 4.67. An equivalent impedance (R-L) replaces the transformers and the distribution lines.
The voltage equation per phase can be written as follows:

\[ v_i = R \cdot i + L \frac{di}{dt} + v_r \]  \hspace{1cm} (4.5.1)

Depend on the input/output specifications for the model; this equation can be implemented in Simulink in two ways as shown in Fig. 4.68.

The first approach is used when the currents are used as an input for the down-stream block while the second one uses the voltage as an output for this block.

The mask interface for these models is shown in Fig. 4.69.

4.5.2 Circuit breaker model

A model for the circuit breaker, which takes the moment when the currents have zero crossing, into account has been implemented in Simulink. The opening mechanism is shown in Fig. 4.70.
The pole will be opened at the first zero-crossing moment for the phase current (e.g. phase B) then based on the zero-crossing moment for the other two phases the remaining poles are opened. Usually, for a 3-phase symmetrical load after the opening of the first pole, the currents through the other two branches are in phase, so these poles are opened in the same moment. The dynamic model of the electric arc has not been taken into account.

The developed Simulink model for the circuit breaker comprises three identical subsystems for each phase (pole) [52]. In Fig. 4.71 is presented the Simulink implementation for one pole of the circuit breaker.

The mask parameters for this block are: opening time, opening resistance, closing time and closing resistance as shown in Fig. 4.72.
4.5.3 Capacitor bank

The model of the capacitor bank is based on a global transfer function derived from the electrical circuit as it is shown in Fig. 4.73. Since in wind turbine applications in series with the capacitor unit is inserted (optional) a detuning reactor, an inductance has been used to model this reactor.

In the continuous domain (s-plane) the transfer function for one branch is given by:

\[
H(s) = \frac{I(s)}{V(s)} = \frac{sC}{s^2 LC + sRC + 1} \quad (4.5.2)
\]

Since the total current can be written as:

\[
I_t(s) = \sum_k I_k(s) \quad (4.5.3)
\]

and the electrical parameters for each branch are identically, the transfer function for \(N\) parallel elements is given by:

\[
H_t(s) = \frac{I_t(s)}{V(s)} = \frac{sNC}{s^2 LC + sRC + 1} \quad (4.5.4)
\]

where \(N\) is the number of parallel branches.

In order to decrease the capacitance value per phase in wind turbine applications usually the capacitors are connected in delta. Therefore (4.5.4) can be used to model the entire capacitor bank.
The Simulink implementation for this model is shown in Fig. 4.74 and the mask interface in Fig. 4.75 respectively.

![Simulink model of a capacitor bank with N parallel branches in delta connection.](image)

Fig. 4.74. Simulink model of a capacitor bank with N parallel branches in delta connection.

![Mask interface for the capacitor bank model](image)

Fig. 4.75. Mask interface for the capacitor bank model

### 4.5.4 Cable model

The equivalent electrical diagram of a cable with the length $\Delta x$ is shown in Fig. 4.76.

![Equivalent electrical diagram per phase for a cable with the length $\Delta x$.](image)

Fig. 4.76. Equivalent electrical diagram per phase for a cable with the length $\Delta x$.

where:

- $R_1$ - longitudinal resistance;
- $L_1$ - longitudinal inductance;
- $C_1$ - transversal capacitance;
- $G_1$ - transversal conductance.

Based on this equivalent circuit the parameters for the steady-state sinusoidal operation can be defined as follows [48]:

76
Wind Turbine Blockset in Matlab Simulink

- Characteristic impedance: 
  \[ Z_c = \frac{R_1 + j\omega L_1}{G_1 + j\omega C_1} = Z_c e^{j\beta} \]

- Complex constant of propagation: 
  \[ \gamma = \sqrt{(R_1 + j\omega L_1)(G_1 + j\omega C_1)} = \alpha + j\beta \]

where \( \alpha \) is the attenuation constant and \( \beta \) is the phase constant. The real part \( \alpha \) corresponds to the attenuation of the direct- and inverse component amplitude for voltage and current.

Based on the values of the cable parameters, four types of models can be defined as follows:

- Cable without losses: \( R_1 \ll \omega L_1, \quad G_1 \ll \omega C_1 \)
- Cable with low losses at high-frequency \( R_1 < \omega L_1, \quad G_1 < \omega C_1 \)
- Cable without distortions \( \frac{R_1}{L_1} = \frac{G_1}{C_1} \)
- Cable with losses at low frequency \( R_1 \gg \omega L_1, \quad G_1 \gg \omega C_1 \)

Usually, in power distribution applications the model for a cable without distortions is used with good results [52]. This model is characterized by the following parameters:

- phase constant \( \beta = \omega \sqrt{L_1 C_1} \);
- phase speed \( v = \frac{1}{\sqrt{L_1 C_1}} \) is independent of frequency;
- attenuation constant \( \alpha = R_1 \sqrt{\frac{C_1}{L_1}} \);
- characteristic impedance \( Z_c = \frac{L_1}{\sqrt{L_1 C_1}}, \quad \varphi_c = 0 \).

The Simulink implementation of this model is shown in Fig. 4.77.

![Fig. 4.77. Cable model in Simulink.](image)

The mask interface for this model is shown in Fig. 4.78.
4.5.5 Transformer model

In this paragraph the dynamic equations for a three-phase two-winding transformer are presented. The model includes the asymmetry of the magnetic circuit as well as the iron losses. The saturation and the hysteresis effect are not taken into account. This model can easily be extended to a three-phase three-winding transformer.

4.5.5.1 3-phase 2-winding transformer

Assuming a linearized model of the core and the surrounding medium, the magnetic circuit of a transformer core may be represented by the following matrix equation [24] and [25]:

\[
\begin{bmatrix}
\Psi \\
\phi
\end{bmatrix} = \begin{bmatrix}
\mathcal{R}
\end{bmatrix}^{-1} \begin{bmatrix}
\phi \\
\mathcal{P}
\end{bmatrix} = \begin{bmatrix}
\mathcal{P}
\end{bmatrix} \begin{bmatrix}
\phi
\end{bmatrix}
\]

(4.5.5)

where:

- $\phi$ - is the applied mmf
- $\Psi$ - is the developed flux
- $\mathcal{R}$ - is the reluctance matrix
- $\mathcal{P}$ - is the inverse reluctance matrix or the permeance matrix

The matrix equation in per-unit for the magnetic circuit is related to the electrical transformer description through the following matrix transformations:

\[
\frac{d}{dt} \begin{bmatrix}
\Psi \\
\phi
\end{bmatrix} = \begin{bmatrix}
\mathcal{P}
\end{bmatrix} \frac{d}{dt} \begin{bmatrix}
\phi
\end{bmatrix}
\]

(4.5.6)

\[
\begin{bmatrix}
\mathcal{N}_i
\end{bmatrix} \frac{d}{dt} \begin{bmatrix}
\Psi
\end{bmatrix} = \begin{bmatrix}
\mathcal{N}_i
\end{bmatrix} \begin{bmatrix}
\mathcal{P}
\end{bmatrix} \begin{bmatrix}
\mathcal{N}_i
\end{bmatrix} \frac{d}{dt} \begin{bmatrix}
I
\end{bmatrix}
\]

where $\mathcal{N}_i$ is a diagonal matrix containing the number of turns of each winding.

It is convenient to separate the permeance matrix into two components: a matrix dealing strictly with ferro-magnetic core, and a matrix containing the terms related to the leakage flux paths:
\[
[V] = [N_1]([P_m] + [P_\sigma]) [N_i] \frac{d[I]}{dt} = [N] [k_m [T_i] + k_\sigma [K]] [N] \frac{d[I]}{dt} \tag{4.5.7}
\]

where:

- \(T_i\) is the ideal transformer matrix, which is a normalized matrix representation of the core
- \(k_m\) is the per-unit mutual inductance of a given phase (e.g. AA)
- \(K\) is a normalized matrix representing the flux leakage paths
- \(k_\sigma\) is the per-unit leakage inductance for a given phase (e.g. A)
- \(N\) is the normalized turns ratio matrix. Formation of the turns ratio matrix \(N\) by normalization of \(N_t\) is important because turns ratios are known to be virtually the same as the voltage ratios in power transformer design, whereas the number of turns for each winding is generally unavailable. When the secondary windings are related to the primary windings the turns ratio matrix \(N\) is equal to unity matrix.

Under the assumption of linear, steady state sinusoidal behaviour, the voltages can be expressed in phasor form:

\[
[V] = j\omega \left( [k_m [T_i] + k_\sigma [K]] [I] \right) = \left( [Z_m] + [Z_\sigma] \right) [I] \tag{4.5.8}
\]

Separation of the transformer description into ideal transformer and leakage matrices is useful in the determination of transformer parameters.

### 4.5.5.2 Determination of ideal transformer matrix

The ideal transformer matrix may be determined from the magnetic circuit of the transformer by standard analysis.

Supplying the various portions of the magnetic circuit with an applied mmf, and calculating the flux incident to all of the transformer windings the elements of these matrices are readily determined. This is similar to the traditional determination of \(Z\)- or \(Y\) parameter models.

For example for a three-leg core the ideal transformer matrix can be calculated based on Fig. 4.79.

![Fig. 4.79. A 3-limb core and the equivalent magnetic circuit.](image)

Applying a unit mmf on each winding and then calculating the flux developed in each leg, the ideal transformer matrix is given by:
Parameter estimation for linear transformer model

For practical system studies, a number of parameters must be obtained in order to generate the primitive (unconnected) transformer models. Measurement of these parameters is a very difficult task for the distribution transformers due to the large number and variety of distribution transformers, thus it is necessary to estimate the transformer parameters.

The most commonly available measurements are series or short-circuit impedances, the no load losses and the magnetizing currents. These measurements and knowledge of the core construction are used to generate transformer models.

The transformer can be described by a impedance matrix which can take into account the winding losses, core losses, core geometry etc.

The steady state voltage equation can then be written as:

\[
\begin{bmatrix}
V \\
\end{bmatrix} = \begin{bmatrix} Z \end{bmatrix} \begin{bmatrix} I \\
\end{bmatrix}
\]

The basic problem is to find the parameters of this impedance matrix.

Determination of parameters for lossless three-phase transformer

The magnetizing currents in the three-phase transformer are determined by application of a balanced 3-phase voltage to transformer primary windings. Since, all other injected currents are zero, the magnetizing currents must satisfy the equation relating the input voltage and the 3x3 submatrix of Z. First only the mutual reactances will be considered for the impedance matrix, the winding resistances and leakage reactances will be neglected.

The magnetizing currents in per unit is given by:
Equation (4.5.11) shows that, the per-unit magnetizing currents are a function only of the core geometry and they are not balanced due to this. Since the manufacturer test data specifies the average magnetizing currents in per-unit it can be defined a scaling factor:

\[ \zeta = k_m = \frac{\left| \frac{I_{U1}}{\text{ave}} \right|}{\omega} \]  

This scaling factor is equal with the per-unit mutual inductance of phase A. The leakage terms may be determined based on the per-unit positive sequence short-circuit reactance since this reactance is the same as the reactance in each of the transformer phases. When only one short-circuit measurement is available for the transformer and no other constraints are known, the reactance may be split evenly between each primary and secondary winding.

**Inclusion of loss terms**

Two types of losses can be included in the transformer model: the core losses associated with hysteresis and eddy current effects and the copper losses associated with windings resistances. These losses can be modeled by adding series resistance terms to the impedance matrix:

\[ Z = R_{\text{winding}} + j\omega k_o [K] + R_{\text{core}} + j\omega k_m [T_i] \]  

In an idealized core model, the core losses may be considered to be equivalent to a distributed set of resistive windings around each core section. The voltage developed across each of these loss windings is proportional to the \( \frac{d\phi}{dt} \) in that section and hence the losses are proportional to the flux distribution in the core and the core geometry. For all practical purposes, the core losses are developed in an equivalent series network given by:

\[ R_{\text{core}} = c [T_i] \]

When the magnetizing current and the core losses are considered the two parameters \( c \) and \( k_m \) must be determined. The magnetizing current is related to these two parameters as:

\[ \zeta = j\omega k_m + c \left| \frac{I_{U1}}{\text{ave}} \right| \]  

The approximation is valid if the windings parameters are very small compared with the core parameters:

\[ R_{\text{winding}} << R_{\text{core}} \]
\[ X_\sigma << X_m \]

Both conditions are satisfied in power distribution transformers.
The core losses can be determined by deriving the equation for the per-unit complex power used in the transformer:

\[ P_{\text{loss}} = \text{Re}\left\{ \left(T_n^2 \right) c \left[T_j^2 \right] \right\} = c \frac{\left| I_{\text{ave}}^2 \right|}{\left| I_{\text{ave}} \right|} \text{Re}\left\{ \left[I_{U}^2 \right] \left[T_j^2 \right] \left[I_{U}^2 \right]^* \right\} = c \frac{\left| I_{\text{ave}}^2 \right|}{\left| I_{U}^2 \right|} P_U = c \frac{1}{\xi^2} P_U \quad (4.5.17) \]

where: \( c = \xi^2 \frac{P_{\text{loss}}}{P_U} \)

The parameter \( k_m \) is defined as:

\[ k_m = \sqrt{\frac{\xi^2 - c^2}{\omega^2}} \]

The conversion of parameters \( c \) and \( k_m \) from per unit values to SI values can be made using the base reactance impedance per phase:

\[ Z_{\text{base}} = \frac{U_{\text{base}}}{I_{\text{base}}} \quad (4.5.18) \]

### 4.5.5.4 Dynamic equations of 3 phase 2 windings transformer

The voltage equations for a 3-phase 2-winding transformer written in natural reference frame (ABC/abc) neglecting saturation are:

\[ [v] = \left\{ [R_w] + [R_{\text{core}}] \right\} [i] + \left\{ [M] + [L_{\text{co}}] \right\} \frac{d[i]}{dt} \quad (4.5.19) \]

where:

\[ [v] = [v_A \ v_B \ v_C \ v_a \ v_b \ v_c]^T \] - are the phase voltages applied to each HV and LV windings;

\[ [i] = [i_A \ i_B \ i_C \ i_a \ i_b \ i_c]^T \] - are the phase currents in each HV and LV windings;

\[ [R_w] = \begin{bmatrix} R_p & R_p & 0 \\ R_p & R_p & 0 \\ 0 & 0 & R_s \end{bmatrix} \] - matrix of windings resistances

\[ [L_{\text{co}}] = \begin{bmatrix} L_{\text{op}} & 0 \\ 0 & L_{\text{op}} \end{bmatrix} \] - matrix of leakage inductances
\[
\begin{align*}
R_{\text{core}} &= cZ_{\text{base}} [T] \\
M &= kZ_{\text{base}} [T]
\end{align*}
\]

Equation (4.5.19) can be implemented in Simulink as shown in Fig. 4.80.

Fig. 4.80. Simulink implementation of the dynamic model for a three-phase two-winding transformer: a) Simulink model and b) mask interface.

Using only parameters from data-sheet the asymmetry of the core as well as the iron losses can be modelled. Moreover, the saturation can easily be added to this model if a complete no-load test is available.

### 4.5.6 Phase Locked Loop

Since a Phase Locked Loop (PLL) is a central component in a control structure for the grid-side converter, it will be presented in this paragraph dedicated to grid components models. The PLL is used to estimate the grid angle and therefore to generate the control signals for the grid side converter. A generalized PLL structure is shown in Fig. 4.81.

The three-phase voltage inputs are transformed into \( \alpha \beta \) components referred to the grid voltage phasor. Using this components sufficient information about the phase angle is.
obtained. A phase detector range of \([-\pi, \pi]\) is obtained using a simple inverse tangent function, while using a four quadrant inverse tangent function this range can be expanded to \([-\pi, \pi]\). The loop filter, contains a low pass filter to suppress noise and high frequency terms in the signal from the phase detector, e.g. due to asymmetry in the three-phase voltages. To avoid stationary error in phase after a step in the input signal frequency, a PI-controller is added. The performance of the PLL depends on the phased detector and the selected bandwidth of the loop filter. With slow dynamic of the PLL loop, higher rejection of the disturbances can be achieved, although a poor tracking of the grid angle.

In the power system applications, the voltage-controlled oscillator usually is implemented as an integrator. The loop is close by feeding the estimate angle into the \(\alpha\beta\)-transformation.

As long as the phase angle is correct, the output of the phase detector is zero, thus the frequency input to the integrator is constant – the PLL is in lock. If the estimated angle is not correct, the frequency is adjusted and the phase angle of the grid voltage phasor is changed.

The Simulink implementation of the PLL is shown in Fig. 4.82.

4.6 Control blocks – Control of PQ for DFIG

This paragraph presents a control scheme for the active and reactive power, which can be used in variable speed wind turbines with doubly fed induction generators.

**Stator-flux oriented control**

In this paragraph the stator flux oriented control will be briefly presented in order to understand the basic principle of this control.

Starting from the voltage equations of the induction machine written in synchronous reference frame the rotor voltage equations will be written as:

\[
\begin{align*}
    v_{rx} &= R_r i_{rx} + L_{rr} \frac{di_{rx}}{dt} + L_{lm} \frac{di_{sx}}{dt} - (\omega_x - \omega_r)(L_{rr} i_{ry} + L_{lm} i_{sy}) \\
    v_{ry} &= R_r i_{ry} + L_{rr} \frac{di_{ry}}{dt} + L_{lm} \frac{di_{sy}}{dt} + (\omega_x - \omega_r)(L_{rr} i_{rx} + L_{lm} i_{sx})
\end{align*}
\]  \hspace{1cm} (4.6.1)

where: \(\omega_x\) is the angular speed of the synchronous reference frame.
The relation between fluxes and currents are given by:

\[
\begin{bmatrix}
\psi_{sx} \\
\psi_{sy} \\
\psi_{rx} \\
\psi_{ry}
\end{bmatrix} =
\begin{bmatrix}
L_{ss} & 0 & L_{m} & 0 \\
0 & L_{ss} & 0 & L_{m} \\
L_{m} & 0 & L_{rr} & 0 \\
0 & L_{m} & 0 & L_{rr}
\end{bmatrix}
\begin{bmatrix}
i_{sx} \\
i_{sy} \\
i_{rx} \\
i_{ry}
\end{bmatrix}
\]

\[\text{(4.6.2)}\]

\[\text{Fig. 4.83. Explicative to stator-flux oriented control.}\]

Attaching the reference frame to the stator flux as shown in Fig. 4.83, the following simplifications can be made:

\[
\psi_{sx} = L_{ss}i_{ms} + L_{m}i_{rx}
\]
\[\psi_{sy} = L_{ss}i_{sy} + L_{m}i_{ry}\]

\[\text{(4.6.3)}\]

where \(i_{ms}\) is the stator magnetizing current.

Based on (4.6.3) the components of the stator current can be expressed as:

\[
i_{sx} = \frac{L_{ms}}{L_{ss}}(i_{ms} - i_{rx})
\]
\[
i_{sy} = -\frac{L_{ms}}{L_{ss}}i_{ry}
\]

\[\text{(4.6.4)}\]

Thus, the governing equations for the rotor currents are:

\[
T_r \frac{di_{sx}}{dt} + i_{sx} = \frac{v_{sx}}{R_r} + (\omega_x - \omega_r)T_i i_{ry} - (\sigma_r - T_r) \frac{di_{ms}}{dt}
\]
\[
T_r \frac{di_{sy}}{dt} + i_{sy} = \frac{v_{sy}}{R_r} + (\omega_x - \omega_r)(T_i i_{rx} + (\sigma_r - T_r)i_{ms})
\]

\[\text{(4.6.5)}\]

where: \(T_r = \frac{D}{R_rL_{ss}}\) is the rotor time constant, and \(\sigma_r = \frac{L_{rr}}{R_r}\).

By inspecting (4.6.5) it can be observed some cross coupling between the x- and y-axis due to the rotational \textit{emf} terms. The current loop dynamic along the two axes can be made independent of each other by compensating these cross coupling terms. However, since the stator windings are directly connected to the grid the stator flux is almost constant in normal operation, the contribution of its derivative can be ignored. Moreover, as the slip range is limited the contribution of these coupling terms is rather weak.
In order to complete the description of the machine behaviour the equation of the active and reactive power at the stator terminals should be added.

Considering that the x-axis component of the stator voltage is zero the general expression of the active and reactive power in synchronous reference frame is:

\[
\begin{align*}
P_s &= \frac{3}{2} v_{sy} i_{sy} \\
Q_s &= \frac{3}{2} v_{sy} i_{sx}
\end{align*}
\]

Taking into account (4.6.4), (4.6.6) can be written as:

\[
\begin{align*}
P_s &= \frac{3}{2} \frac{L_m}{L_{ss}} v_{sy} i_{sy} \\
Q_s &= \frac{3}{2} v_{sy} \left( \frac{L_m}{L_{ss}} i_{ms} - i_{sx} \right)
\end{align*}
\]

Usually, the rotor controller design is based on (4.6.5), while the control of active and reactive power is based on (4.6.5)-(4.6.7). This control can be implemented in Simulink as shown in Fig. 4.84.

![Fig. 4.84. Control of active and reactive power for a doubly fed induction generator implemented in Simulink.](image)

The parameters for this control block are determined from the equivalent scheme parameters of the induction machine (standard data-sheets).

This control can be used both for a complete dynamic model of the induction machine or for a reduced order one.
4.7 Summary

This chapter presents the mathematical models for the main components within a wind turbine system, namely: wind model, aerodynamic model, drive train model, generator models, power converters and modulation strategies, transformers, cable model, etc.

All these models have been implemented in Matlab/Simulink and collected into a Toolbox dedicated to wind turbine applications.

Since this Toolbox comprises much more models, some of the mathematical models, which are not so relevant, have not been presented here. In Appendix B other models are described in terms of Simulink diagram, mask interface and input parameters.
Chapter 5

Simulation results

Using the developed models from Wind Turbine Blockset from Matlab/Simulink some analysis have been performed for the main concepts used in wind turbine systems: namely: fixed-speed wind turbines and variable speed/pitch wind turbines with induction generators. The simulations present an analysis for the connection and disconnection of a three-phase transformer used in fixed speed-wind turbines. Then the start-up sequence of a squirrel-cage induction machine using a soft-starter is shown. Finally, the control of active and reactive power for a variable pitch/speed wind turbine with doubly fed induction generator is shown.

5.1 Connection of a no-load transformer to the grid

The moment when a transformer used in a fixed speed wind turbine is connected to the grid has been simulated. For this analysis the new developed model for the three-phase transformer has been used. The main goal in this analysis was to study the influence of the main cable between transformer and grid-connection point.

For simulations it has been assumed that there is no load on the secondary side (Low Voltage side) of the transformer as shown in Fig. 5.1.

The Simulink diagram comprises the transformer model, the cable model and the grid model as shown in Fig. 5.2.

The simulated phase-to-phase and phase-to-ground voltages are presented in Fig. 5.3.
Notice in Fig. 5.3 the harmonic content of the voltages during the switch on transients. These oscillations are due to the main cable between the grid and the transformer.

5.2 Disconnection of a no-load transformer

The moment when a three-phase transformer is disconnected by the main switchgear from the grid without load on the secondary side (LV side) has been investigated. The analysis uses only the developed models from Wind Turbine Blockset. In this case the influence of the cable between transformer and switchgear has been studied. During the disconnection an equivalent capacitance including cable capacitance and the transformer parasitic capacitances are taken into account. The equivalent diagram of the system in this case is shown in Fig. 5.4.

The Simulink diagram for this system is shown in Fig. 5.5.
The simulated phase-to-phase and phase-to-ground voltages are presented in Fig. 5.6.

![Transformer phase-ground voltages](image)

![Transformer phase-phase voltages](image)

Fig. 5.6. Transformer voltages during disconnection from grid:
- a) phase-ground voltages
- b) phase-phase voltages

It can be observed in Fig. 5.6 the harmonic content of the phase-to-ground voltages due to the presence of the parasitic capacitances and the capacitance of the cable between the circuit breaker and the transformer. These harmonic are suppressed in the line voltages due to the connection type of the High-Voltage windings (delta connection in this case).

### 5.3 Start-up sequence of a fixed-speed wind turbine

The start-up sequence of a soft-starter-fed squirrel-cage induction machine, which is used in wind turbine applications, has been studied. The induction machine has 2 MW rated power, 690 V / 1700 A rated phase-voltage and rated line current respectively (delta connection). The induction machine is connected via soft-starter to the supply voltage below synchronous speed (1450 rpm). The starting firing angle for the soft-starter is 120°.

The equivalent diagram of this system is shown in Fig. 5.7.

![Equivalent diagram of the system](image)

Fig. 5.7. Equivalent diagram of the system.

This system has been modelled using the available blocks from *Wind Turbine Blockset* as shown in Fig. 5.8.
In order to evaluate such a system during the start-up sequence the electromagnetic torque and the rotational speed at the high-speed shaft are analysed in two cases. These variables are shown in Fig. 5.9 in case of a direct-start-up sequence and using a soft-starter.

When the induction machine is connected directly to the grid high starting torque values are recorded as well as a high harmonic content (50 Hz). Large oscillations in the shaft speed are also present. Using a soft-starter the inrush currents and therefore the high-starting torque are limited and the shaft speed is smooth.

In order to highlight the different operation modes of the soft-starter during the start-up sequence the phase voltage and the corresponding line current for different firing angles are shown in Fig. 5.10.
Fig. 5.10. Phase voltage and line current for different firing angles during the start-up sequence:
a) Mode 3 of operation and b) Mode 1 of operation.

These waveforms can be obtained using only an $ABC/abc$ model for the induction machine. Moreover, the $ABC/abc$ developed model from the Wind Turbine Blockset permits to analyse both phase and line voltages and currents.

5.4 Variable speed/pitch wind turbine with DFIG

Using the available blocks from the “Wind Turbine Blockset” the control of active and reactive power for a 2 MW wind turbine using doubly fed induction generator (DFIG) has been studied. The Simulink diagram of the system is shown in Fig. 5.11.

The simulation structure comprises the wind model, drive train model, a DFIG model written is synchronous reference frame, the control block for active and reactive power (P&Q) and the optimal control of the entire system. The optimal control block implements basically the same control strategy as in DIgSILENT. The algorithm used in the P&Q control block can be used with a reduced model for DFIG. Therefore, the main goal of this simulation was to test the P&Q control algorithm for a future implementation in HAWC. Notice that the power converter has been omitted in this simulation because an average model of it is taken into account in the P&Q control block.
In order to analyse the control of the active and reactive power for this system a wind time series with an average value of 10 m/sec has been used. The synchronous speed of the machine has been considered as the base value for speed, while the rated power of the machine is the base for the active and reactive power.

The simulations results in terms of the wind time series, active and reactive power, both for the stator and the rotor circuit, are shown in Fig. 5.12.

![Simulation results for a variable speed/pitch wind turbine with DFIG.](image)

It can be observed that the power is limited at the rated value, while the speed is lower than the rated value. The reference of the stator reactive power and the measured one are zero in the entire simulation horizon. Since, the wind speed has been acquired with the inherited sample time from simulation (0.05 sec) and used in the control algorithm, the reference for the stator power is not so smooth and the produced active power follows identically this reference. Into a real system the wind speed is acquired with a bigger sample time and some calculation in order to find the average wind speed at each 1 min are performed. Due to this filtering of the wind speed the reference is much more smooth and the output power will not exhibit this fast variations. It has been omitted the averaging block of the wind speed in order to study the dynamics performances of the control loops.

### 5.5 Summary

Using the developed models from *Wind Turbine Blockset* from Matlab/Simulink some analysis have been performed for the main concepts used in wind turbine systems, namely: fixed-speed wind turbines and variable speed/pitch wind turbines with induction generators. The developed models can be used in a wide range of analysis for wind turbine systems.
Conclusions and Future work

Matlab/Simulink is a very powerful tool for developing new models and new control algorithms. In principle any physical system can be modelled based on a mathematical description of it. However, a wind turbine is a very complex system and the interconnection of these subsystems requires many assumptions and modelling “artifices”. Even if the models are optimized for a high-speed simulation, the “real” computational time for a complete wind turbine system is large. Depends on the modelling aspects and time horizon some simplifications in the model are required in order to be able to simulate such a system.

A Matlab/Simulink Toolbox for wind turbine applications has been developed during the Simulation Platform Project. This toolbox contains models for the components from a wind turbine system. Basically, the main wind turbine concepts namely: fixed-speed and variable speed/pitch wind turbines can be simulated using the developed models.

Wind turbine systems contain subsystems with different ranges of the time constants: wind, turbine, generator, power electronics, transformer and grid. Among these components the electrical generators and the power converters need the smallest simulation time step and therefore, these blocks decide the simulation speed.

Much attention has been on increasing the simulation speed for all models from this Toolbox. The same algorithm can be implemented in Simulink in different ways. For example an algebraic or differential equation can be implemented using elementary blocks or just using a Function block. Dependent on the implementation method used in Simulink the simulation time can be significantly reduced. Different methods of building a model have been tested and it has been found that:

- The number of blocks used in a model has a big influence on the total simulation time. This number can be reduced using the matrix support and some special blocks from Simulink libraries, e.g. Function Block;
- Using a C S-Function instead of using Simulink blocks to implement a model with a large number of derivatives the total simulation time can be further reduced;
- Using a C S-Function the algebraic loops, which are inevitable in a large and complex model can be avoided. The algebraic loops in a Simulink model are iteratively computed at every time step. Therefore, they severely degrade the performances and the total simulation time. The main problem here is that Simulink uses at this moment only Ordinary Differential Equation Solvers (ODE). In order to handle mathematical systems, which involves both differential equations and algebraic equations a Differential and Algebraic Equation Solver (DAE) should be used.

Some of the features of this Toolbox can be summarized as follows:

- All the developed models use basically only Simulink Blocks. No particular Toolbox is necessary to simulate a wind turbine concept. So, the user should have installed on the PC only Matlab and Simulink. However, some measurement block uses particular blocks from other Toolboxes from Matlab e.g. Power Spectra Density calculation block;
- It uses the matrix support in order to minimize the number of blocks and connection lines;
- All models which involve a great number of differential equations (e.g. electrical machines, drive-trains and transformer) are available also as ‘C’ S-Functions for high-speed simulations;

- In order to be able to use different drive-train models the equation of motion is not included in the electrical machine models;

Special models have been developed for induction generators (ABC/abc c models), which can be used in some particular fault analysis e.g. unsymmetrical, unbalanced faults with unbalanced loads. In this case cannot be used the standard dqq model. Moreover, this model is used to simulate the soft-starter fed induction machines, which are used in fixed-speed wind turbines.

The developed models have been verified and validated in some research projects or particular analysis (e.g. Middelgrunden wind farm).

Currently, some wind turbine manufacturers evaluate this toolbox e.g. Vestas, KK-Electronic A/S. Some models are used by other companies, universities or have been used in student projects (Master) or PhD projects. Also there are requests from universities all over the world (Spain, China, Brazil, etc.).

The work during the project for developing these new models in Matlab/Simulink constitutes the base of several papers published at international conferences and journals.

In order to extend the capabilities of this Toolbox to simulate and analyse wind turbine/farm topologies some general guidelines for future work are as follows:

- To include saturation and iron losses in all electrical machine models based on standard data-sheets;
- To develop a 3-phase 3-winding model for transformer;
- To include saturation in the transformer models;
- To develop Abc/abc models for synchronous machine both field winding and permanent magnet;
- To develop an advanced model for overhead-lines and shielded cables;
- To develop an Differential and Algebraic Equation Solver (DAE) for Simulink;
- To develop models for new topologies and components, which can be used in future in the wind turbine applications e.g. switched reluctance generator, matrix converter, new control strategies for wind turbine/farm, etc.

A continuous update of the existing models from this toolbox will be made based on the feedbacks from users.

Updates for the Wind Turbine Blockset to the current Matlab version will be necessary. In the last year Mathworks Inc. has performed two major updates for the basic products (Matlab 6.5 and Matlab 6.5 ServicePack 1) and each time a review of the developed models from the Wind Turbine Blockset has imposed.

In December 2003 Mathworks Inc. has announced that a new version of Matlab (Matlab 7) will be released in January 2004. It is expected that this new release will contain major changes again.
References

[7]. Bala, I. – Electrical Machines, EDP, Bucharest 1982;
[18]. Data, R., Ranganathan, V.T. – Decoupled control of active and reactive power for a grid connected doubly-fed wound rotor induction machine without position sensor, pp. 2623-2630;
Wind Turbine Blockset in Matlab Simulink


[38] Leonhard, W. - Control of electric drives, Springer Verlag, 2001, ISBN 3540418202;


[54] Pillay, P., Levin, V. - Mathematical models for induction machines;

Wind Turbine Blockset in Matlab Simulink

[64] Sallan, J., Muljadi, E., Sanz, M. – Control of Self-Excited Induction Generators Driven by Wind, Proceedings of EPE 99, Laussane, Switzerland;
[70] Smiai, M.S., Lesan, S., Shepherd, W. – Analysis and performance of hybrid thyristor controlled three-phase resistive inductive loads,
[77] Yamamoto, M., Motoyoshi, O. – Active and reactive power control for doubly-fed wound rotor induction generator;
[80] *** - Using Control System Toolbox, Mathworks Inc, 2000;
[83] *** - Power Factory, DigSilent, User Manuals;
[84] *** - PSCAD, User Manuals;
[85] *** - Saber, Synopsis, User Manuals;
Appendix A

An example of a “C” S-Function

/* === Dynamic model of Squirrel Cage Induction Machine === */
/* === DQ - rotor reference frame === */
/* Last revision 15.05.2002 */
/* Florin Iov */
#define S_FUNCTION_NAME dq_scim
#define S_FUNCTION_LEVEL 2

#include "simstruc.h"
#include "mex.h"
#include "c:\matlabR12\sys\lcc\include\math.h"
#include "c:\matlabR12\sys\lcc\include\stdlib.h"

#define U(element) (*uPtrs[element]) /* Pointer to Input PortO */
#define PI 3.14159265358979323846
#define c 0.66666666666666666666
#define NPARAM 5 /* number of parameters in mask */
#define NSTATES 5 /* number of continuous states */
#define NINPUT 4 /* number of inputs */
#define NOUTPUT 4 /* number of outputs */
#define NRWORK 3 /* number of work pointers */

#define STATOR_PARAM(S) ssGetSFcnParam(S,0)
#define ROTOR_PARAM(S) ssGetSFcnParam(S,1)
#define MAG_PARAM(S) ssGetSFcnParam(S,2)
#define NUMB_POLES(S) ssGetSFcnParam(S,3)
#define X0_PARAM(S) ssGetSFcnParam(S,4)

/* S-function methods */
static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumSFcnParams(S, NPARAM); /* Number of expected parameters */
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S))
    {
        /* Return if number of expected != number of actual parameters */
        return;
    }

    ssSetNumContStates(S, NSTATES);
    ssSetNumDiscStates(S, 0);

    if (!ssSetNumInputPorts(S, 1)) return;
    ssSetInputPortWidth(S, 0, NINPUT);
    ssSetInputPortDirectFeedThrough(S, 0, 0);
if (ssSetNumOutputPorts(S, 1)) return;
ssSetOutputPortWidth(S, 0, NOUTPUT);

ssSetNumSampleTimes(S, 1);
ssSetNumRWork(S, NRWORK);
ssSetNumIWork(S, 0);
ssSetNumPWork(S, 0);
ssSetNumModes(S, 0);
ssSetNumNonsampledZCs(S, 0);

ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE);
}

/* Function: mdlInitializeSampleTimes =========================================*/
static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0);
}

/*========================================================*/
#define MDL_INITIALIZE_CONDITIONS

static void mdlInitializeConditions(SimStruct *S)
{
    real_T *x0 = ssGetContStates(S);
    real_T *workPointer0 = ssGetRWork(S);
    int_T i;
    int_T m;

    if (inxGetM(X0_PARAM(S)) != 0)
    {
        const real_T *pr = inxGetPr(X0_PARAM(S));

        for (i = 0; i < NSTATES; i++)
        {
            *x0++ = *pr++;
        }
    }
    else
    {
        for (i = 0; i < NSTATES; i++)
        {
            *x0++ = 0.0;
        }
    }

    for (m = 0; m < 3; m++)
    {
        *workPointer0++ = 0;
    }
}

/* Function: mdlOutputs =======================================================*/
static void mdlOutputs(SimStruct *S, int_T tid)
{

real_T *y = ssGetOutputPortRealSignal(S,0);
real_T *x = ssGetContStates(S);

const real_T *Lm = mxGetPr(MAG_PARAM(S));
const real_T *p = mxGetPr(NUMB_POLES(S));

y[0] = x[0] * cos(x[4]) - x[1] * sin(x[4]); /* isa */
y[3] = 1.5 * *p * *Lm * (x[1] * x[2] - x[0] * x[3]); /* torque */

/*========================================================*/
#define MDL UPDATE
static void mdlUpdate(SimStruct *S, int_T tid)
{
    real_T *x = ssGetContStates(S);
    InputRealPtrsType uPtrs = ssGetInputPortRealSignalPtrs(S,0);
    real_T *workPointer = ssGetRWork(S);
    const real_T *p = mxGetPr(NUMB_POLES(S));

    workPointer[0] = (c * ( U(0) * cos(x[4]) + U(1) * cos(x[4] - c * PI) + U(2) * cos(x[4] + c * PI) )); /* vds */
    workPointer[1] = -(c * ( U(0) * sin(x[4]) + U(1) * sin(x[4] - c * PI) + U(2) * sin(x[4] + c * PI) )); /* vqs */
    workPointer[2] = *p * U(3); /* omega rotor */
}
/*========================================================*/
#define MDL_DERIVATIVES
static void mdlDerivatives(SimStruct *S)
{
    real_T dx = ssGetDX(S);
    real_T *x = ssGetContStates(S);
    InputRealPtrsType uPtrs = ssGetInputPortRealSignalPtrs(S,0);
    const real_T *workPointer = ssGetRWork(S);

    const real_T *stator = mxGetPr(STATOR_PARAM(S));
    const real_T *rotor = mxGetPr(ROTOR_PARAM(S));
    const real_T *Lm  = mxGetPr(MAG_PARAM(S));

    real_T Rs, Lsgms, Rr, Lsgmr, Ls, Lr, sigma;
    real_T a11, a12, a13, a14, a15;
    real_T a31, a32, a33, a34, a35;

    Rs = stator[0]; /* stator resistance */
    Lsgms = stator[1]; /* stator leakage inductance */
    Rr = rotor[0]; /* rotor resistance */
    Lsgmr = rotor[1]; /* rotor leakage inductance */

    Ls = Lsgms + *Lm; /* stator self inductance */
    Lr = Lsgmr + *Lm; /* rotor self inductance */

    sigma = 1 - (*Lm * *Lm) / (Ls * Lr); /* leakage coefficient */

    /* matrix coefficients */
    a11 = Rs / (sigma * Ls);
\[a_{12} = \frac{1}{\sigma};\]
\[a_{13} = (R_r \cdot L_m)/(L_s \cdot L_r \cdot \sigma);\]
\[a_{14} = \frac{L_m}{(L_s \cdot \sigma)};\]
\[a_{15} = \frac{1}{(L_s \cdot \sigma)};\]
\[a_{31} = (R_s \cdot L_m)/(L_s \cdot L_r \cdot \sigma);\]
\[a_{32} = \frac{L_m}{(L_r \cdot \sigma)};\]
\[a_{33} = \frac{R_r}{(\sigma \cdot L_r)};\]
\[a_{34} = \frac{L_m \cdot L_m}{(L_s \cdot L_r \cdot \sigma)};\]
\[a_{35} = \frac{L_m}{(L_s \cdot L_r \cdot \sigma)};\]

/* derivates */
\[dx[0] = -a_{11} \cdot x[0] + \text{workPointer}[2] \cdot a_{12} \cdot x[1] + a_{13} \cdot x[2] + a_{14} \cdot \text{workPointer}[2] \cdot x[3] + a_{15} \cdot \text{workPointer}[0]; /* ids */\]
\[dx[1] = -a_{12} \cdot \text{workPointer}[2] \cdot x[0] - a_{11} \cdot x[1] - a_{14} \cdot \text{workPointer}[2] \cdot x[2] + a_{13} \cdot x[3] + a_{15} \cdot \text{workPointer}[1]; /* iq */\]
\[dx[2] = a_{31} \cdot x[0] - a_{32} \cdot \text{workPointer}[2] \cdot x[1] - a_{33} \cdot x[2] - a_{34} \cdot \text{workPointer}[2] \cdot x[3] - a_{35} \cdot \text{workPointer}[0]; /* idr */\]
\[dx[3] = a_{32} \cdot \text{workPointer}[2] \cdot x[0] + a_{31} \cdot x[1] + a_{34} \cdot \text{workPointer}[2] \cdot x[2] - a_{33} \cdot x[3] - a_{35} \cdot \text{workPointer}[1]; /* iqr */\]
\[dx[4] = \text{workPointer}[2]; /* theta */\]

/* Function: mdlTerminate */
static void mdlTerminate(SimStruct *S)
{
    UNUSED_ARG(S); /* unused input argument */
}

/*=============================================*
 /* Required S-function trailer */
*=============================================*/

#ifdef MATLAB_MEX_FILE  /* Is this file being compiled as a MEX-file? */
#include "simulink.c" /* MEX-file interface mechanism */
#else
#include "cg_sfun.h" /* Code generation registration function */
#endif
Appendix B

Other developed models

Harmonic voltage source and grid angle detection

A voltage source with a harmonic content according with the EN61000-2-2 standard has been built in Matlab/Simulink. This standard “set” the harmonic content in the utility grid as shown in Table B.1.

<table>
<thead>
<tr>
<th>Harmonic number</th>
<th>1st</th>
<th>3rd</th>
<th>5th</th>
<th>7th</th>
<th>9th</th>
<th>11th</th>
<th>13th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>100%</td>
<td>5%</td>
<td>6%</td>
<td>5%</td>
<td>1.5%</td>
<td>3.5%</td>
<td>3%</td>
</tr>
<tr>
<td>Phase</td>
<td>178.2°</td>
<td>0°</td>
<td>0°</td>
<td>0°</td>
<td>0°</td>
<td>0°</td>
<td>0°</td>
</tr>
</tbody>
</table>

The Simulink implementation of this harmonic voltage source is shown in Fig. B.1 and Fig. B.2.

The mask interface offers the possibility to select the desired harmonic content as shown in Fig. B.3.
Using this voltage source the Phase Locked Loop model has been tested for different jumps in amplitude, frequency and phase of the grid voltage.

The simulation results in terms of the “real” voltages given by the Harmonic Voltage Source and the reconstructed voltages using the estimated grid angle from the PLL are shown in Fig. B.4.

In Fig. B.4 should be notice the triple zero crossing for the “real” phase voltages. The simulations have been performed for severe jumps in the amplitude, frequency and phase angle of the Harmonic Voltage Source. The absolute error between the “real” grid angle and the estimated one is shown in Fig. B.5. The PLL block can estimate the grid angle accurately for severe jumps in the amplitude and frequency of the grid voltages. The response of the PLL is still good for jumps in the phase angle of the grid voltages.
Wind Turbine Blockset in Matlab Simulink

Fig. B.5. Absolute error angle between the “real” grid angle and the estimated angle from the PLL block.

**Three-phase diode bridge rectifier**

![Three-phase diode bridge rectifier diagram](image)

Fig. B.6. Simulink implementation of the three-phase diode bridge rectifier.

**Synchronous PWM modulator**

This modulation strategy for power converters is implemented in Simulink as shown in Fig. B.7. The model synchronizes the carrier frequency of the modulator with the fundamental frequency based on the frequency modulation index. This index has a fixed value in this case. However, in order to have the frequency modulation index as an input parameter the mask interface shows in Fig. B.7 can be modified.
ABC to Magnitude and phase transformation

This Transformation block converts a three-phase symmetrical system (voltages or currents) into the RMS value, phase and zero sequence of it. The Simulink implementation is shown in Fig. B.8.

Average wind calculator

This measurement block calculate the average wind speed value for a given time interval. The Simulink model shows in Fig. B.9 comprises a special blocks from DSP Blockset namely Mean calculator.