Titre: Title:	Electromagnetic transient modeling of asynchronous machine in modelica, accuracy and performance assessment
Auteurs: Authors:	Alireza Masoom, & Jean Mahseredjian
Date:	2024
Type:	Article de revue / Article
Référence: Citation:	Masoom, A., & Mahseredjian, J. (2024). Electromagnetic transient modeling of asynchronous machine in modelica, accuracy and performance assessment. IEEE Access, 12, 135864-135872. https://doi.org/10.1109/access.2024.3462255

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/59445/
Version:	Version officielle de l'éditeur / Published version Révisé par les pairs / Refereed
Conditions d'utilisation: Terms of Use:	CC BY-NC-ND

Titre de la revue: Journal Title:	IEEE Access (vol. 12)
Maison d'édition: Publisher:	Institute of Electrical and Electronics Engineers
URL officiel: Official URL:	https://doi.org/10.1109/access.2024.3462255
Mention légale: Legal notice:	This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 July 2024, accepted 12 September 2024, date of publication 16 September 2024,
date of current version 30 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3462255

RESEARCH ARTICLE

Electromagnetic Transient Modeling of Asynchronous Machine in Modelica, Accuracy, and Performance Assessment

ALIREZA MASOOM[1], (Member, IEEE), AND JEAN MAHSEREDJIAN[2], (Life Fellow, IEEE)
[1]Hydro-Québec Research Institute, Varennes, QC J3X 1S1, Canada
[2]Department of Electrical Engineering, Polytechnique Montréal, Montreal, QC H3T 1J4, Canada

Corresponding author: Alireza Masoom (masoom.alireza@hydroquebec.com)

This work was supported by NSERC, Hydro-Québec, RTE, EDF, and OPAL-RT as a part of the industrial chair "Multi Time-Frame Simulation of Transients for Large Scale Power Systems."

ABSTRACT Classical EMT-type simulators are mostly programmed in procedural languages, e.g. Fortran or C. In these languages, the focus is mainly on the solution methods. Modern languages, such as Modelica, are declarative and primarily focused on modeling and simulation. Modelica offers a much higher abstraction level, which makes the codes more concise and understandable. This paper contributes to the electromagnetic transient modeling and simulation of asynchronous machines in Modelica. In this paper, the modeling of a three-phase squirrel cage (single and double cage) and wound-rotor induction machine in three different reference frames is described and implemented. The accuracy and performance of Modelica models are compared and validated with the classical modeling approach used in the reference software EMTP. It is demonstrated that Modelica-based models with variable-step solvers offer fast and accurate results for time-domain simulations of motor sequential startup cases.

INDEX TERMS Asynchronous machine, sequential start-up, Modelica, equation-based modeling, variable-step solver.

I. INTRODUCTION

Electromagnetic Transient (EMT) modeling and simulation deal with power electric components and systems in the form of differential equations for the study of a wideband range of phenomena such as switching, lightning, ferroresonace, and electromechanical oscillations [1]. The commonly used EMT-detailed models are implemented either using the block-diagram programming approach, e.g., Simscape Electrical Specialized Power System [2], or using an *imperative* language such as Fortran and C++ in, for example, EMTP® [3]. The block-diagram approach is at a higher level, and easily accessible, but the modeler must define the flow of information and the sequence of blocks.

Asynchronous machines (ASM) are widely used in power systems in the form of motors or generators in renewable energy sources, such as the Doubly Fed Induction Generator. Modeling of ASMs is principally based on the selectable reference frame. In the classical modeling approach, first, the differential equations of the ASM are discretized using a fixed-step solver, then using the companion circuit, the model is represented through a Norton equivalent (NE) in modified-augmented-nodal analysis (MANA) in [3], for example. The NE is based on a given numerical integration technique, such as commonly used trapezoidal integration, for example. In [3], it is optionally possible to iterate with network equations to achieve higher accuracy. Although such a solution method is fast and suitable for large circuits, it relies on a specific solver and detailed coding of all equations and procedures. Spurious oscillations can be suppressed using a damping resistor in parallel with inductors in the machine model [4]. In the classical approach, the controllers are modeled separately from the main electrical network. In some simulation packages [2], the machine models are

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Zhao.

implemented using a current source. This approach may typically cause numerical instability problems, especially with larger time steps. To preserve simulation stability, a resistance whose value is dependent on the simulation time-step can be connected to the machine terminals [2].

Modelica [5] is an object-oriented *declarative* language. The modeler's concentration is on model equations rather than giving a stepwise algorithm on how to achieve the desired goal. In Modelica, a model can be explicitly described by its Differential-Algebraic Equations and block diagrams at a much higher abstraction level and most importantly, it is independent of the underlying solution method. Modelica can be linked to various EMT-type codes through FMI [6], [7]. Modelica supports a graphical user interface (GUI) for designing the model symbol and drawing the network diagram. It simplifies the component connectivity checks, documentation, and observing the results as well. Several integration methods, such as the trapezoidal, backward Euler, DASSL [8], and IDA [9] are currently available in Modelica simulators, e.g. OpenModelica [10] and Dymola [11].

In Modelica, each component is coded by a specialized class `model` or `block` (can be used in the block diagram approach). Other types of classes exist to support modeling, including `function` (to provide algorithmic contents), `package` (to provide a structuring mechanism), `connector` (to define the model terminals), and `partial model` (used as base). In the class `model`, each electrical component is represented by the parameters, variables, and finally, the relation between variables of a model regardless of which one is known or unknown. During the compiling of Modelica codes, a flattening procedure is run to map the hierarchical structure of a model into a set of differential, algebraic and discrete equations. The solution direction of equations will adapt to the data flow dependencies between the equations. The analysis is not just sorting but also manipulation of equations, which includes, for example, index reduction and tearing of equations, to convert the coefficient matrix into block lower triangular form.

The Modelica language is frequently used in phasor domain simulations [12], for voltage and transient stability applications [13]. OpenIMDML [14], for example is a specialized library for simulations of drives and inductions machines in phasor domain. The existing machine models in the Modelica Standard Library [15] provide only basic models with several limitations and do not allow to perform EMT-type detailed simulations. The authors of this paper recently considered the Modelica language for EMT simulations [16], [17] including power electronics [18]. In addition, some other efforts have been carried out to accelerate the simulations using data locality and vectorization techniques [19] and compiled models in Dynawo [20].

This paper focuses on EMT-detailed asynchronous machine models (single and double cages, wound-rotor) development in Modelica. It is demonstrated that the modern approach helps modelers concentrate on modeling rather than the solution methods. In the numerical examples, the performance and accuracy of ASM models are tested during fault and network islanding. This paper also presents the advantages of variable-step solvers for the simulation and analysis of motor sequential start-ups. It is also demonstrated that simulation converges directly with different tolerances and without the addition of artificial components.

This paper is organized as follows. In Section II, the theory of ASM models is described. In Section III, the implementation of the models in Modelica is presented. The models are validated using three examples in Section IV.

II. ASYNCHRONOUS MACHINE THEOROTICAL DESCRPTION

A. WOUND ROTOR AND SINGLE SQUIRREL CAGE MACHINES

A single squirrel cage machine is widely used as a motor in industry and can be represented as a wound-rotor machine in which the terminals are short-circuited. Therefore, the model of the wound-rotor machine will be discussed in this section. Fig. 1 shows the q-axis equivalent circuit of a wound-rotor ASM. The following equations describe the *abc* to reference frame transformations applied for ASM [21].

$$\mathbf{v}_{qds} = \mathbf{P}(\theta)\,\mathbf{v}_{abcs} \tag{1}$$

$$\mathbf{v}_{qdr} = \mathbf{P}(\beta)\,\mathbf{v}_{abcr} \tag{2}$$

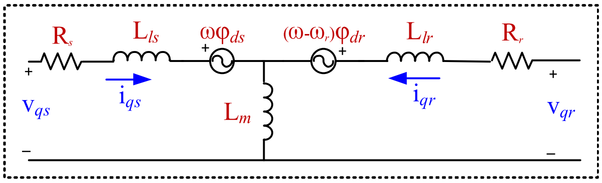


FIGURE 1. The q-axis equivalent circuit of a wound-rotor ASM in time-domain.

where $\mathbf{P}$ denotes the Park transformation matrix, θ and $\beta = \theta - \theta_r$ represent the angular positions of the reference frame and the difference between the positions of the reference frame and the rotor position, θ_r, respectively. Depending on the chosen reference frame, θ can take different values; for example, if the rotor reference frame is chosen, then $\theta = \theta_r$.

Electrical equations of ASM can be declared by selecting the current or flux as independent variables, but from the simulation standpoint, if flux is chosen as the state variable, in the voltage equations only one derivative of flux appears, which causes more efficient computations. Equation (3) represents the electrical behavior of an ASM by a fourth-order *nonlinear* flux state-space model in per-unit (pu) form. In the following equations, bold uppercase represents matrices, bold lowercase denotes vectors and p is the derivative operator.

$$p\boldsymbol{\varphi} = \omega_{base}\left(\mathbf{A}\boldsymbol{\varphi} + \mathbf{v}\right) \tag{3}$$

$$\mathbf{A} = -\left(\mathbf{R}\mathbf{L}^{-1} + \mathbf{W}\right) \tag{4}$$

$$\mathbf{i} = \mathbf{L}^{-1}\boldsymbol{\varphi} \tag{5}$$

$$\mathbf{i}_{abcs} = \mathbf{P}^{-1}(\theta)\,\mathbf{i}_{qds} \tag{6}$$

$$\mathbf{i}_{abcr} = \mathbf{P}^{-1}(\beta)\,\mathbf{i}_{qdr} \tag{7}$$

where

$$\boldsymbol{\varphi} = \begin{bmatrix} \varphi_{qs} & \varphi_{ds} & \varphi_{qr} & \varphi_{dr} \end{bmatrix}^{\mathrm{T}} \tag{8}$$

$$\mathbf{v} = \begin{bmatrix} \mathrm{v}_{qs} & \mathrm{v}_{ds} & \mathrm{v}_{qr} & \mathrm{v}_{dr} \end{bmatrix}^{\mathrm{T}} \tag{9}$$

$$\mathbf{i} = \begin{bmatrix} \mathrm{i}_{qs} & \mathrm{i}_{ds} & \mathrm{i}_{qr} & \mathrm{i}_{dr} \end{bmatrix}^{\mathrm{T}} \tag{10}$$

$$\mathbf{R} = diag(\mathrm{R}_s, \mathrm{R}_s, \mathrm{R}_r, \mathrm{R}_r) \tag{11}$$

$$\mathbf{L} = \begin{bmatrix} \mathrm{L}_{ls}+\mathrm{L}_m & 0 & \mathrm{L}_m & 0 \\ 0 & \mathrm{L}_{ls}+\mathrm{L}_m & 0 & \mathrm{L}_m \\ \mathrm{L}_m & 0 & \mathrm{L}_{lpr}+\mathrm{L}_m & 0 \\ 0 & \mathrm{L}_m & 0 & \mathrm{L}_{lpr}+\mathrm{L}_m \end{bmatrix} \tag{12}$$

$$\mathbf{W} = \begin{bmatrix} 0 & \omega & 0 & 0 \\ -\omega & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega-\omega_r \\ 0 & 0 & -(\omega-\omega_r) & 0 \end{bmatrix} \tag{13}$$

In the above equations, all parameters and variables are computed on the stator side and the indices s and r represent the stator and rotor, respectively. The variables $\mathbf{v}_{abcs}$ and $\mathbf{v}_{abcr}$ are the stator and rotor terminal voltages; the vectors $\mathbf{v}$, $\mathbf{i}$, and $\boldsymbol{\varphi}$ denote the stator and rotor voltages; currents, and flux linkages in the qd frame, $\mathbf{i}_{abcs}$ and $\mathbf{i}_{abcr}$ are the stator and rotor current vectors. R_s and R_r are stator and rotor resistances. $\mathbf{L}$ is a symmetrical matrix, and L_{ls}, L_{lpr} and L_m indicates the stator and rotor leakage inductances and mutual inductance respectively. In (13), for the rotor reference frame, $\omega = \omega_r$, for stator reference frame $\omega = 0$ and finally for synchronous reference frame, $\omega = 1$.

The mechanical equations in pu that describe the dynamics of the single mass rotor are given as:

$$\mathrm{T}_e = \varphi_{ds}\mathrm{i}_{qs} - \varphi_{qs}\mathrm{i}_{ds} \tag{14}$$

$$p\omega_r = \frac{1}{2\mathrm{H}}(\mathrm{T}_e - \mathrm{T}_m) \tag{15}$$

$$p\theta_r = \omega_{base}\omega_r \tag{16}$$

where T_e represents the electromagnetic torque, ω_{base} is the base electrical angular velocity in radian per second and H represents the inertia constant in pu-s. It is noted the mechanical equations can be easily extended to a multi-mass system.

For single-cage ASM, the terminals of rotor windings are short-circuited, therefore, (9) is given as follows. Other equations remain intact.

$$\mathbf{v} = \begin{bmatrix} \mathrm{v}_{qs} & \mathrm{v}_{ds} & 0 & 0 \end{bmatrix}^{\mathrm{T}} \tag{17}$$

B. DOUBLE-CAGE INDUCTION MOTOR

Double-cage induction motors are used in applications where high starting torque is required. This arrangement increases the rotor resistance which is directly proportional to the starting torque of an induction motor. (9) Fig. 2 shows the q-axis equivalent circuit of a double-cage rotor for time-domain solution. The equivalent circuit is based on EMTP and differs from other EMT-type simulation packages, where the second cage is represented by duplicating and parallel connection of the single-cage rotor model [23]. Likewise, the same circuit is obtained for d-axis by replacing the indices and reversing the polarity of speed voltage sources.

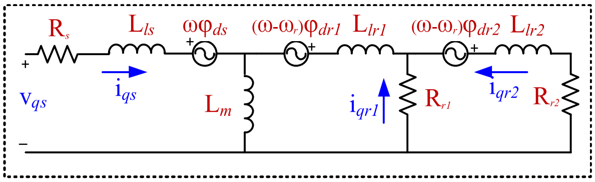


FIGURE 2. The q-axis equivalent circuit of a double-cage-rotor in time-domain.

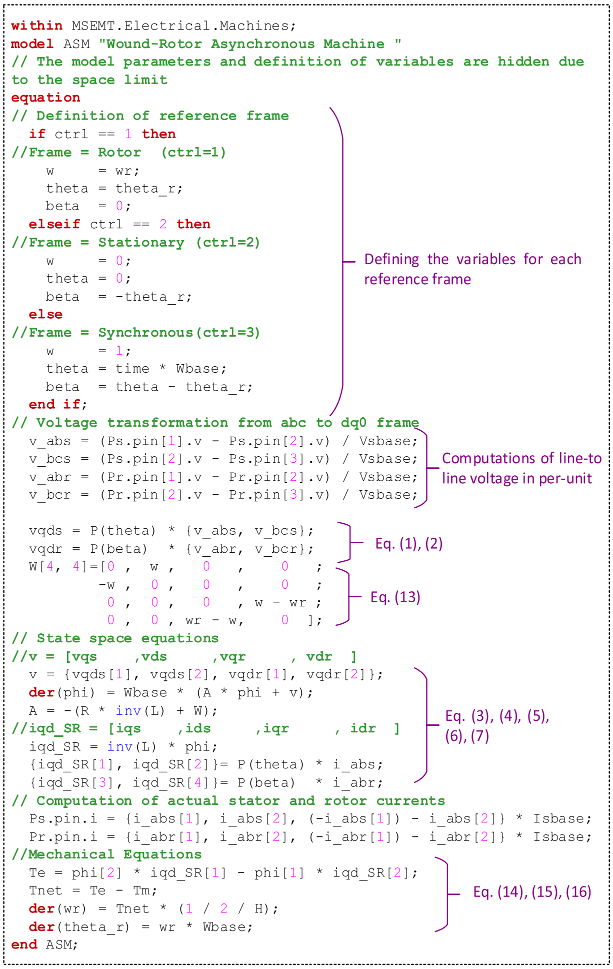
Equations (1)-(7) and (14)-(16) remain valid for the double-cage machines, Equations (8)-(13) are re-defined as (18)-(23), shown at the bottom of the next page.

In the above equations, indices $r1$ and $r2$ represent the properties of cage 1 and 2 respectively. In (23), $\mathrm{L}_{lr1}$and L_{lr2} are the leakage inductances of cages 1 and 2, respectively.

III. MODEL IMPLEMENTATION IN MODELICA

This section presents the modeling of wound-rotor ASM in Modelica. Fig. 3 shows the high-level Modelica codes for the implementation of associated equations. The definition of variables and parameters is hidden due to space limitations and only the `equation` compartment is shown. In these pieces of code, firstly, the rotating reference is selected. The stator and rotor winding terminals are respectively represented by `Ps` and `Pr`. In Modelica, the physical terminal is coded as `connector` class and is composed of 3 pins (`pin` [1], `pin` [2], and `pin` [3]). For each `pin`, two variables; voltage (denoted by `v`), and current (denoted by `i`), are defined. Current is characterized as a `flow` variable. When two pins are connected, it means the connector variables without the `flow` prefix are equal (i.e. `v`) and the sum of corresponding connector variables with the `flow` prefix (i.e. `i`) equals zero. Then, using the selected reference frame, the stator and rotor voltage vector, denoted respectively by `vqds` and `vqdr` are computed. The Park transform is coded in a separate class (i.e. `function`) and denoted by `P`. The general rotation matrix,$\mathbf{W}$ is coded as per (13). The elements of this matrix are time-variant, and change based on the selected reference frame. Then, the state space electrical equation of ASM is implemented. In this equation, flux, `phi`, is a vector of state variables and defined as per (3). Stator and rotor current vector `iqd_SR`, is calculated using (5). Equation (10) defines the elements of this vector. Then, the stator and rotor currents are converted to abc frame.

Computation of electromagnetic torque and other electromechanical equations are coded as per (14)-(16). One can see that Modelica offers high-level and understandable codes in a few lines that match the equations of the model. In this approach, the equations are defined by "=," and the unknowns can lie on both sides, and there is no pre-defined causality (input/output data flow). There is no constraint to connect the terminals of the machine to the voltage or current source. The simulation engine must reorder the equations symbolically to determine their execution order.



```
within MSEMT.Electrical.Machines;
model ASM "Wound-Rotor Asynchronous Machine "
// The model parameters and definition of variables are hidden due
to the space limit
equation
// Definition of reference frame
  if ctrl == 1 then
//Frame = Rotor  (ctrl=1)
    w     = wr;
    theta = theta_r;
    beta  = 0;
  elseif ctrl == 2 then
//Frame = Stationary (ctrl=2)
    w     = 0;
    theta = 0;
    beta  = -theta_r;
  else
//Frame = Synchronous(ctrl=3)
    w     = 1;
    theta = time * Wbase;
    beta  = theta - theta_r;
  end if;
// Voltage transformation from abc to dq0 frame
  v_abs = (Ps.pin[1].v - Ps.pin[2].v) / Vsbase;
  v_bcs = (Ps.pin[2].v - Ps.pin[3].v) / Vsbase;
  v_abr = (Pr.pin[1].v - Pr.pin[2].v) / Vsbase;
  v_bcr = (Pr.pin[2].v - Pr.pin[3].v) / Vsbase;

  vqds = P(theta) * {v_abs, v_bcs};
  vqdr = P(beta)  * {v_abr, v_bcr};
  W[4, 4]=[0 ,  w ,    0   ,    0    ;
          -w ,  0 ,    0   ,    0    ;
           0 ,  0 ,    0   , w - wr ;
           0 ,  0 , wr - w,    0    ];
// State space equations
//v = [vqs    ,vds     ,vqr     , vdr  ]
  v = {vqds[1], vqds[2], vqdr[1], vqdr[2]};
  der(phi) = Wbase * (A * phi + v);
  A = -(R * inv(L) + W);
//iqd_SR = [iqs    ,ids     ,iqr     , idr  ]
  iqd_SR = inv(L) * phi;
  {iqd_SR[1], iqd_SR[2]}= P(theta) * i_abs;
  {iqd_SR[3], iqd_SR[4]}= P(beta)  * i_abr;
// Computation of actual stator and rotor currents
  Ps.pin.i = {i_abs[1], i_abs[2], (-i_abs[1]) - i_abs[2]} * Isbase;
  Pr.pin.i = {i_abr[1], i_abr[2], (-i_abr[1]) - i_abr[2]} * Isbase;
//Mechanical Equations
  Te = phi[2] * iqd_SR[1] - phi[1] * iqd_SR[2];
  Tnet = Te - Tm;
  der(wr) = Tnet * (1 / 2 / H);
  der(theta_r) = wr * Wbase;
end ASM;
```

Defining the variables for each reference frame

Computations of line-to line voltage in per-unit

Eq. (1), (2)

Eq. (13)

Eq. (3), (4), (5), (6), (7)

Eq. (14), (15), (16)

FIGURE 3. Modelica code for implementation of wound-rotor ASM.

Fig. 4 shows the pieces of code for modeling of double-cage ASM. One can see that the modeling of double-cage ASM is easily achieved by minimum manipulations in the existing codes (outline on this figure). For example, the matrices **W** is redefined as (22), the pin `Pr` is removed, and the voltage vector is modified as per (19). The current vector should be changed according to (20). Eq. (7) is re-used for computation of current of the second cage. The electrical equations (3)-(5) and mechanical ones (14)-(16) are remained intact.

Fig. 5. shows the icon and GUI designed for wound-rotor ASM. One can see the electrical pins (i.e. `Ps` and `Pr`) and control pins (e.g. `Tm`) for electrical and control connection. The GUI connectivity check prevents the user from connecting the electrical and control signal pins together wrongly. In addition, Modelica provides the properties to categorize the model parameters, and to change the units of parameters. Also, a graphical interface is available to plot variables from simulated models.

IV. NUMERICAL TESTS AND VALIDATION

In this section, three numerical case studies are presented to compare the accuracy, performance, and numerical stability of Modelica ASM modeling with EMTP. All simulations are performed in OpenModelica, an open-source Modelica compiler in ODE mode.

A. CASE STUDY 1, MODEL VALIDATION

Fig. 6 shows a circuit case for the model validation of the single-cage ASM during startup and short-circuit events [23]. The machine is a 900 hp, 2.4 kV, 4-pole motor with full-load power factor of 88 %, full-load current of 202 A, stator resistance of 0.08 Ω, rotor resistance of 0.09 Ω, stator leakage reactance of 0.32 Ω, rotor leakage reactance at locked rotor of 0.45 Ω; the magnetizing reactance is 16 Ω, full load slip is 0.15%, and inertia constant H is 1 s. The mechanical torque of the motor is modeled using the *nonlinear* block Fu,

$$\boldsymbol{\varphi} = \left[\varphi_{qs}\ \varphi_{ds}\ \varphi_{qr1}\ \varphi_{dr1}\ \varphi_{qr2}\ \varphi_{dr2} \right]^{\mathrm{T}} \tag{18}$$

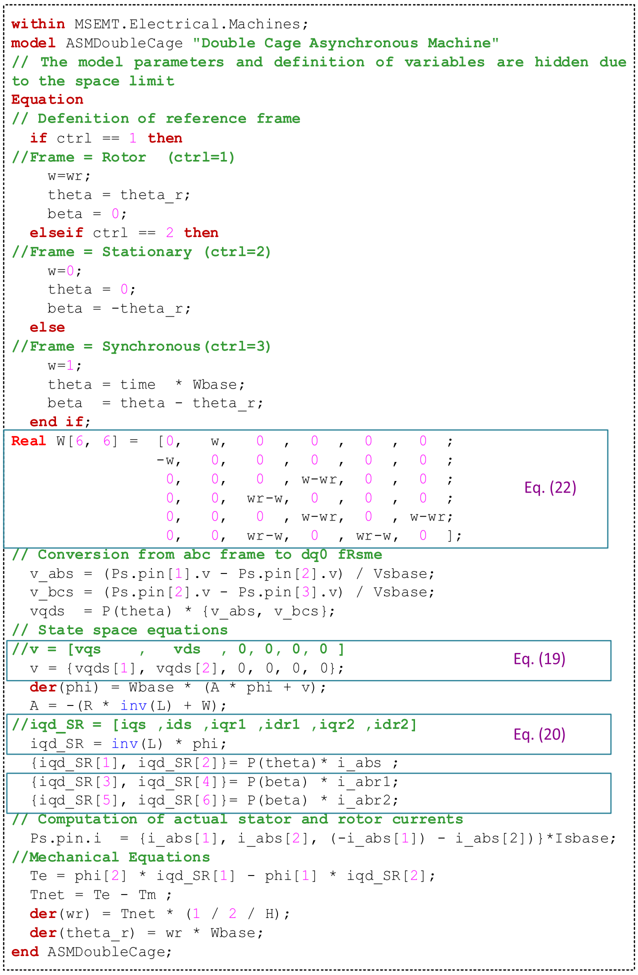
$$\mathbf{v} = \left[\mathrm{v}_{qs}\ \mathrm{v}_{ds}\ 0\ 0\ 0\ 0 \right]^{\mathrm{T}} \tag{19}$$

$$\mathbf{i} = \left[\mathrm{i}_{qs}\ \mathrm{i}_{ds}\ \mathrm{i}_{qr1}\ \mathrm{i}_{dr1}\ \mathrm{i}_{qr2}\ \mathrm{i}_{dr2} \right]^{\mathrm{T}} \tag{20}$$

$$\mathbf{R} = \mathrm{diag}\left(\mathrm{R}_s, \mathrm{R}_s, \mathrm{R}_{r1}, \mathrm{R}_{r1}, \mathrm{R}_{r2}, \mathrm{R}_{r2}\right) \tag{21}$$

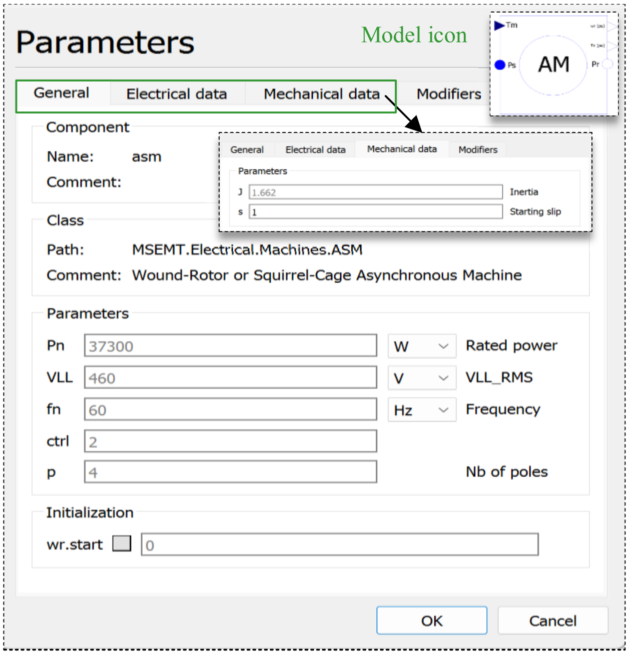
$$\mathbf{W} = \begin{bmatrix} 0 & \omega & 0 & 0 & 0 & 0 \\ -\omega & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega - \omega_r & 0 & 0 \\ 0 & 0 & -(\omega - \omega_r) & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega - \omega_r & 0 & \omega - \omega_r \\ 0 & 0 & -(\omega - \omega_r) & 0 & -(\omega - \omega_r) & 0 \end{bmatrix} \tag{22}$$

$$\mathbf{L} = \begin{bmatrix} \mathrm{L}_{ls} + \mathrm{L}_m & 0 & \mathrm{L}_m & 0 & \mathrm{L}_m & 0 \\ 0 & \mathrm{L}_{ls} + \mathrm{L}_m & 0 & \mathrm{L}_m & 0 & \mathrm{L}_m \\ \mathrm{L}_m & 0 & \mathrm{L}_{b1} + \mathrm{L}_m & 0 & \mathrm{L}_{b11} + \mathrm{L}_m & 0 \\ 0 & \mathrm{L}_m & 0 & \mathrm{L}_{br1} + \mathrm{L}_m & 0 & \mathrm{L}_{br1} + \mathrm{L}_m \\ \mathrm{L}_m & 0 & \mathrm{L}_{br1} + \mathrm{L}_m & 0 & \mathrm{L}_{br1} + \mathrm{L}_{br2} + \mathrm{L}_m & 0 \\ 0 & \mathrm{L}_m & 0 & \mathrm{L}_{br1} + \mathrm{L}_m & 0 & \mathrm{L}_{ir1} + \mathrm{L}_{lr2} + \mathrm{L}_m \end{bmatrix} \tag{23}$$



```
within MSEMT.Electrical.Machines;
model ASMDoubleCage "Double Cage Asynchronous Machine"
// The model parameters and definition of variables are hidden due
to the space limit
Equation
// Defenition of reference frame
  if ctrl == 1 then
//Frame = Rotor   (ctrl=1)
    w=wr;
    theta = theta_r;
    beta = 0;
  elseif ctrl == 2 then
//Frame = Stationary (ctrl=2)
    w=0;
    theta = 0;
    beta = -theta_r;
  else
//Frame = Synchronous(ctrl=3)
    w=1;
    theta = time  * Wbase;
    beta  = theta - theta_r;
  end if;
Real W[6, 6] =  [0,   w,   0  ,  0  ,  0  ,  0  ;
                -w,   0,   0  ,  0  ,  0  ,  0  ;
                 0,   0,   0  , w-wr,  0  ,  0  ;
                 0,   0,  wr-w,  0  ,  0  ,  0  ;
                 0,   0,   0  , w-wr,  0  , w-wr;
                 0,   0,  wr-w,  0  , wr-w,  0  ];
// Conversion from abc frame to dq0 fRsme
  v_abs = (Ps.pin[1].v - Ps.pin[2].v) / Vsbase;
  v_bcs = (Ps.pin[2].v - Ps.pin[3].v) / Vsbase;
  vqds  = P(theta) * {v_abs, v_bcs};
// State space equations
//v = [vqs    ,   vds  , 0, 0, 0, 0 ]
  v = {vqds[1], vqds[2], 0, 0, 0, 0};
  der(phi) = Wbase * (A * phi + v);
  A = -(R * inv(L) + W);
//iqd_SR = [iqs ,ids ,iqr1 ,idr1 ,iqr2 ,idr2]
  iqd_SR = inv(L) * phi;
  {iqd_SR[1], iqd_SR[2]}= P(theta)* i_abs ;
  {iqd_SR[3], iqd_SR[4]}= P(beta) * i_abr1;
  {iqd_SR[5], iqd_SR[6]}= P(beta) * i_abr2;
// Computation of actual stator and rotor currents
  Ps.pin.i  = {i_abs[1], i_abs[2], (-i_abs[1]) - i_abs[2])}*Isbase;
//Mechanical Equations
  Te = phi[2] * iqd_SR[1] - phi[1] * iqd_SR[2];
  Tnet = Te - Tm ;
  der(wr) = Tnet * (1 / 2 / H);
  der(theta_r) = wr * Wbase;
end ASMDoubleCage;
```

FIGURE 4. Modelica code for implementation of double cage ASM.



Parameters

Model icon

General | Electrical data | Mechanical data | Modifiers

Component

Name: asm

Comment:

General | Electrical data | Mechanical data | Modifiers

Parameters

J 1.662 Inertia

s 1 Starting slip

Class

Path: MSEMT.Electrical.Machines.ASM

Comment: Wound-Rotor or Squirrel-Cage Asynchronous Machine

Parameters

Pn	37300	W	Rated power
VLL	460	V	VLL_RMS
fn	60	Hz	Frequency
ctrl	2		
p	4		Nb of poles

Initialization

wr.start □ 0

OK | Cancel

FIGURE 5. The GUI of wound-rotor ASM.

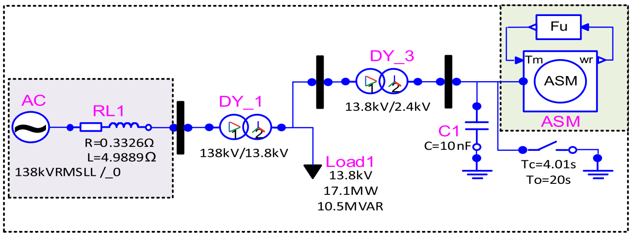


FIGURE 6. System configuration for starting and terminal short-circuit of a 900-hp motor.

whose input is rotor angular velocity. A three-phase short-circuit occurs at t=4.01 s. The reference frame is rotating synchronously. The simulation is run for 5 s with a tolerance of 1e-6 using the DASSL solver [8]. The same circuit is simulated with a time-step of 10 μs in EMTP using the trapezoidal-backward Euler (TBE) integration method.

Fig. 7 compares the different simulation results between Modelica and EMTP (distinguished by black curves). Fig. 7.(a) compares the stator current of phase-a. It is observed that the transient results match perfectly in both solutions. The motor starting current is 10 pu and decays at t=1.5 s. It is observed in Fig. 7.(b) that at this time, the rotor transient currents decay highly, and the rotor accelerates to its rated speed. The results match in this figure as well. At t=4.01 s, a three-phase fault occurs, and a decaying transient appears in both stator and rotor currents. The frequency of rotor transients is nearly 60 Hz since the rotor speed is nearly 1 pu. It is seen that both stator and rotor transients are damped quickly due to stator and rotor flux interactions. Both solutions are identical during the transients.

Fig. 7.(c, d and e) show and compare the solutions of both simulators for the rotor angular velocity, slip and torque. It is seen that the motor starts from stall and reaches steady-state speed (at t=1.5 s) of 0.98 pu. The steady-state slip of the motor is 1.09%. It is observed that in all plots the results obtained from Modelica solution match the ones from EMTP. Fig. 8 illustrates the relative errors between EMTP and Modelica solution points for phase-a stator current. The red and blue curves indicate the errors for the tolerances of 1e-6 and 1e-3, respectively. The norms of relative error for red and blue curves are 0.64% and 0.77%, respectively. It is seen that halving the solver tolerance has a slight impact on the precision of Modelica solutions.

Table 1 compares the runtimes between Modelica and EMTP. It is observed that increasing the tolerance effectively decreases the CPU time with DASSL. One can see that the EMTP outperforms the Modelica with a slight difference.

B. CASE STUDY 2

Fig. 9.(a) shows a more complicated network with two synchronous machines and six double-cage ASMs [22]. The schematic diagram is designed in OpenModelica using the components of the MSEMT library [16]. Each ASM is a

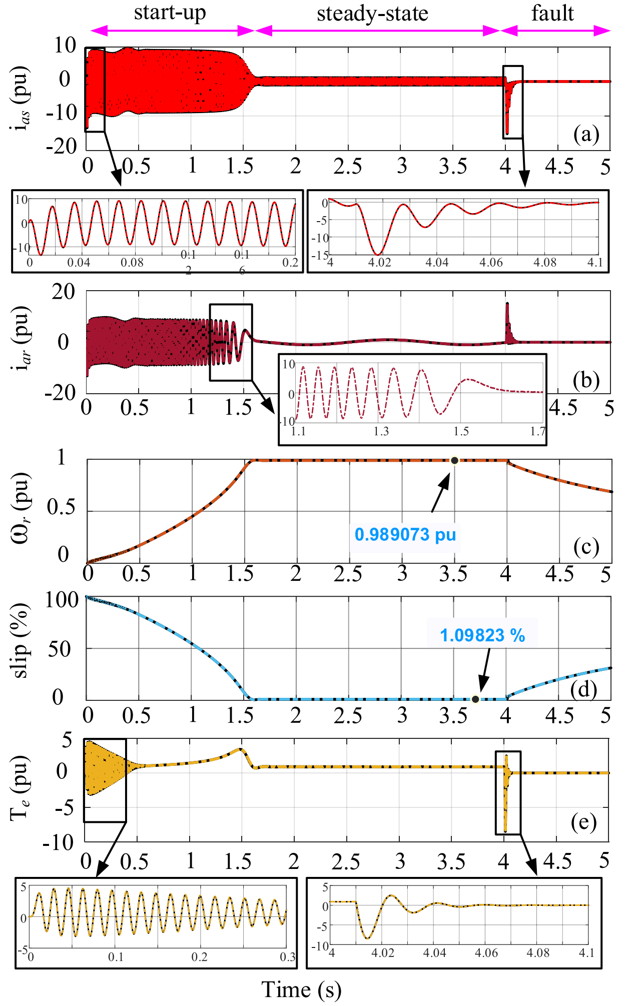


FIGURE 7. Transients of a 900-hp induction motor during start-up, steadystate and three-phase fault at its terminals. EMTP results are distinguished by black curves. (a): stator current phase-*a* (b): rotor current phase-*a* (c) rotor angular velocity (d) slip (e) electromagnetic torque.

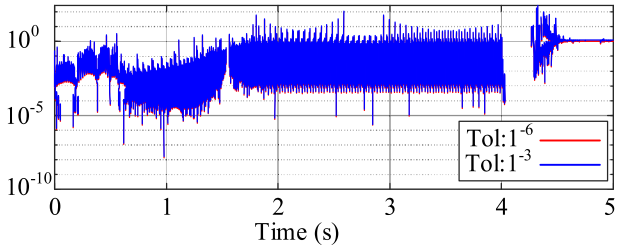


FIGURE 8. Relative error (logarithmic scale) of stator current phase-*a* with different tolerances.

4 pole, 6.6 kV, 1100 hp motor. The breaker (SW_ASM1) is initially open and closes at time $t = 1$ s. This initiates the motor ASM1 startup. Fig. 9.(b) illustrates a nonlinear load model that defines the input mechanical torque of ASM1 as a function of rotor speed ($T_m \propto \omega^2$). The other five induction motors (ASM2 to ASM6) are simultaneously started at $t = 0$ s. The synchronous reference frame is used in all ASMs.

TABLE 1. Comparison of runtime between modelica and EMTP.

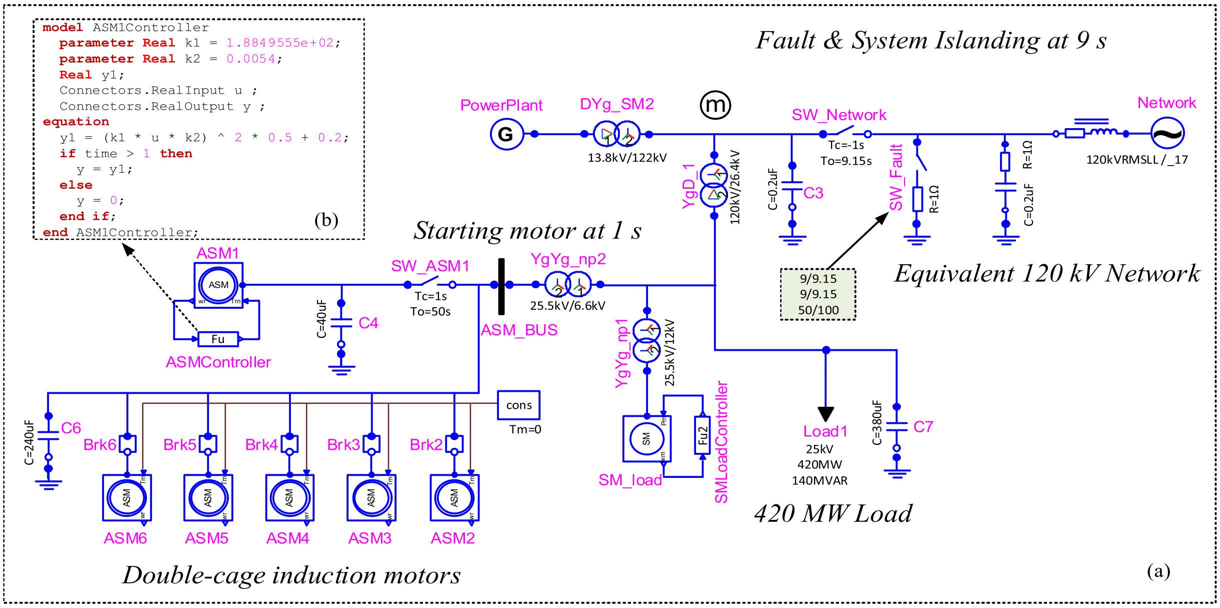
Simulator	Modelica		EMTP
Solver	DASSL		Trap/BE
Tolerance	1e-6	1e-3	-
Timestep (Δt)	-	-	10 µs
No. solution pts	227 280	168 850	500 003
CPU time (s)	7.06 s	5.38	4.42 s

A two-phase-to-ground fault occurs at $t = 9$ s on the 120 kV system and is cleared at $t = 9.15$ s (9 cycles), causing the islanding of the network (the switch SW_Network opens at $t = 9.15$ s). In this network, there are four transformers modeled using independent single-phase units with the nonlinear magnetizing branch. The generator, PowerPlant, consists of detailed models of a 13.8 kV, 500 MVA synchronous machine, exciter (type ST1A), and governor (type IEEEG1). The PowerPlant connects to the network through the transformer DYg_SM2. Simulation is run for 15 s with a tolerance of 1e-6 in OpenModelica, and in EMTP using the TBE solver with a time-step of 200 μs. The simulation in Modelica does not start from steady-state.

Fig. 10 compares the voltage waveforms obtained from both simulators at the interconnection point (point "m" on Fig. 9.(a)). As one can see in Fig. 10. (a), the voltage reaches 1 pu when the system is in steady-state. When the fault occurs, the voltage decreases suddenly to 0.5 pu in the faulty phases.

Fig. 10.(b) shows the close-up view of the voltage waveform of phase-*a* after removing the fault and supplying the system only through PowerPlant. The high-frequency transients are observed in the red curve. The solutions of EMTP depicted by the black curve do not capture these transients using the time-step of 200 μs. This is because Modelica benefits from a variable-step solver in which the time-step is controlled automatically by the rate of transients in the network. Fig. 10.(c) shows the phase-*a* voltage close-up curve during steady-state, where both Modelica and EMTP solution points are identical. It is observed that the synchronous machine, PowerPlant, does not lose its synchronism and the islanded network preserves its stability.

Fig. 11.(a, b) compare the solution points of both simulators for phase-*a* stator current of ASM1. At the instant (t=1 s) the switch SW_ASM1 closes and a large starting current of 5 pu, is drawn by the machine till t=9.15 s. When the network is islanded and supplied only through PowerPlant, the inrush current increases to 7 pu. Fig. 11.(c) shows rotor angular velocity. The speed builds up and finally settles at a speed slightly lower than the synchronous speed at t=14 s. At this moment, the current passing through the stator of ASM1 decreases to 1 pu. Fig. 11.(d) plots the transient electromagnetic torque of ASM1. The torque signal fluctuates at 60 Hz, instantly after energizing the motor due to the transient offset of stator currents. The fluctuations decay at t=3 s to reach 0.5 pu. The electromagnetic torque reaches zero during the fault. After removing the fault and isolating the network, the governor and exciter of PowerPlant are trying to regulate



```
model ASM1Controller
  parameter Real k1 = 1.8849555e+02;
  parameter Real k2 = 0.0054;
  Real y1;
  Connectors.RealInput u ;
  Connectors.RealOutput y ;
equation
  y1 = (k1 * u * k2) ^ 2 * 0.5 + 0.2;
  if time > 1 then
    y = y1;
  else
    y = 0;
  end if;
end ASM1Controller;
```

FIGURE 9. Case study 2: (a) Modelica network, 6 double-cage ASMs and 2 synchronous machines; (b) nonlinear function defining the mechanical torque versus rotor speed.

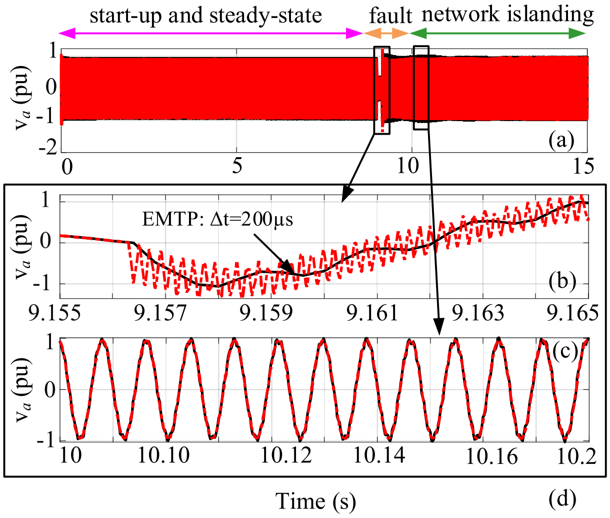


FIGURE 10. Voltage waveform of phase-*a* at point m, Modelica solutions in red curves, EMTP solutions in black curves.

the network voltage and power, therefore the electromagnetic torque starts increasing from $t=10$ s and reaches steady-state at $t=14$ s. It is observed that in all cases the solutions of Modelica and EMTP are identical.

Fig. 12 compares the free-acceleration electromagnetic torque-speed characteristics of ASM6. It is observed that the electromagnetic torque reaches the steady-state value of 0.4 pu when the rotor speed is nearly 0.07 pu. It is once again observed that the solution points of Modelica match EMTP.

Table 2 and Fig. 13 compare the runtimes and results between Modelica and EMTP for different precisions. In EMTP, using the time-step of 20 μs yields results similar to Modelica, whereas the 200 μs time-step does not capture the faster transitions. Decreasing the time-step to 1 μs in EMTP

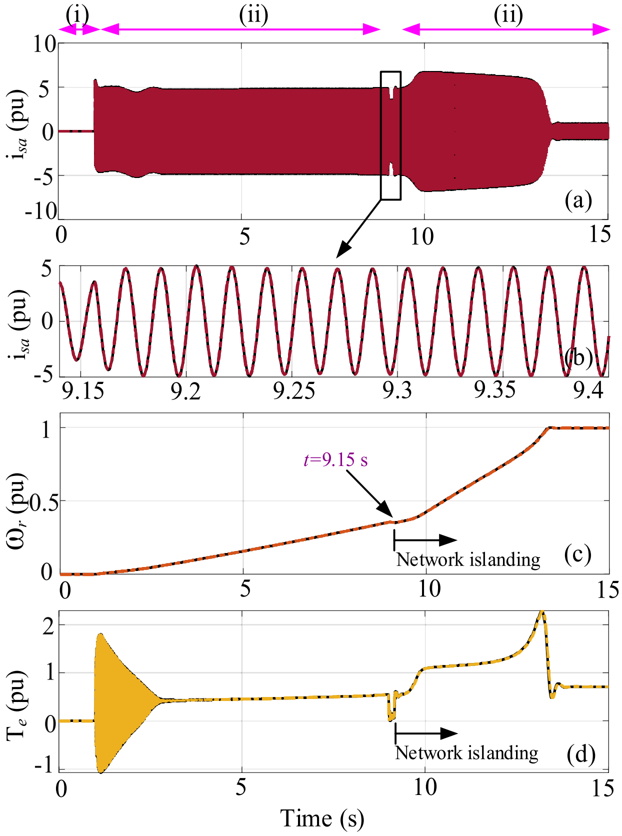


FIGURE 11. Transient curves of ASM1 in three stages (i): stall (ii): from stall to fault. (iii): post- fault and islanding the network.; (a): stator current phase-*a*, (b): zoom-in view of phase-*a*, (c): rotor angular velocity, (d): electromagnetic torque.

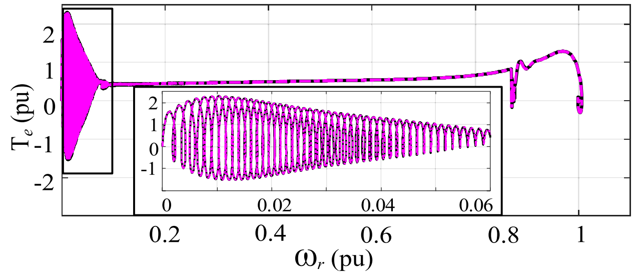


FIGURE 12. Torque-speed characteristic of ASM6. EMTP solution points are distinguished by the black curve.

TABLE 2. Voltage curve of phase-*a* at bus ASM during the sequential start of ASM2 to ASM6.

Simulator	MSEMT		EMTP		
Solver	DASSL		Trap/BE		
Tolerance	1e-6	1e-3	-	-	-
Timestep (Δt)	-	-	200 µs	20 µs	1 µs
No. time points	996362	582689	75008	750009	15000009
CPU time (s)	23.54 s	12.4 s	6.22 s	44.26 s	1119 s

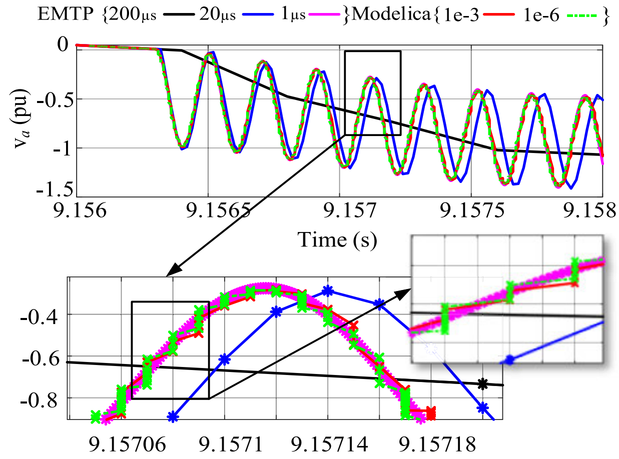


FIGURE 13. Voltage curve at point m, for different resolutions obtained from EMTP and Modelica solvers.

results in the closest solutions to Modelica; in this case, the simulation time increases to 1119 s.

In Modelica, increasing the tolerance decreases CPU time, but it does not show a significant effect on the waveforms. In this case, when transitioning from transients to steady-state, over a long simulation period, the variable-step solver offers better performance than the fixed-step solver of EMTP. The average time-step sizes for the tolerances of 1e-3 and 1e-6 in Modelica are 2.5452e-05 and 1.5052e-05 respectively.

C. CASE STUDY 3, SEQUENTIAL MOTOR STARTUPS

This case aims to exploit further the advantages of a variable time-step solver used in Modelica for the simulation of motor sequential startups in a practical plant simulation case.

A group of motors ASM2 to ASM5 starts sequentially in order from $t = 15$ s to $t = 30$ s with equal delays of 5 s and ASM6 starts at $t = 50$ s, after damping down all transients resulting from previous startups. Simulation is run for 60 s with the time-step of 20 μs in EMTP to capture faster transients. Similarly, the tolerance DASSL is chosen 1e-3 to yield accurate results. Fig. 14 illustrates the phase-*a* voltage waveform of ASM_BUS. It is seen that the results are identical to EMTP. The simulations last 112.6 s and 198.4 s respectively in Modelica and EMTP, which indicates the better performance of variable-step solver for this type of simulation, i.e. sequential startup. Increasing the delays between sequential start-ups increases the performance of Modelica as well. The average step size in the Modelica solver is 168.9 μs.

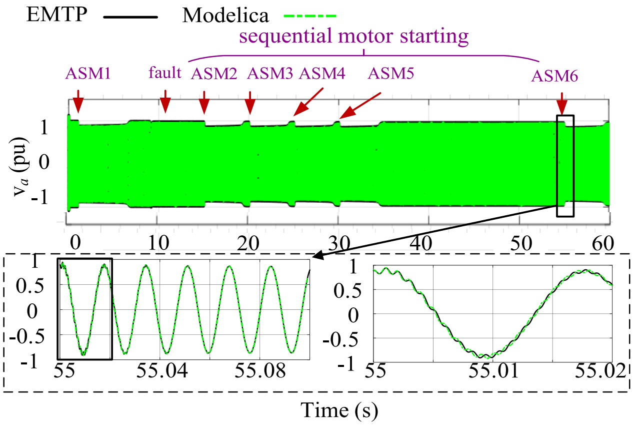


FIGURE 14. Voltage curve of phase-*a* at Bus ASM during the sequential start of ASM2 to ASM6.

V. CONCLUSION

This paper contributed a modern approach for the detailed electromagnetic modeling and simulation of asynchronous machines (ASMs) using Modelica. The electrical and mechanical equations of the machine are developed through a nonlinear state-space model for different *qd*-reference frames. Then, these equations are explicitly implemented using high-level, declarative, equation-based constructs. It was demonstrated that the proposed models are implemented in a few lines of code, are simply modifiable, expandable, and highly legible. The formulation of models is explicitly based on their true mathematical equations. Modelica simulations remain stable without the addition of artificial damping resistances.

It is shown that Modelica can achieve comparable numerical performance with EMTP for the presented ASM simulations. In fact, for the application of motor startup simulations over a long simulation interval, the variable time-step solver used in Modelica allows to achieve better performance. The advantages of variable time-step solution are highlighted, and research may follow with such implementations in EMTP.

Further research is being conducted for using Modelica for large-scale power systems with inverter-based resources.

REFERENCES

[1] J. Mahseredjian, V. Dinavahi, and J. A. Martinez, "Simulation tools for electromagnetic transients in power systems: Overview and challenges," *IEEE Trans. Power Del.*, vol. 24, no. 3, pp. 1657–1669, Jul. 2009, doi: 10.1109/TPWRD.2008.2008480.

[2] (Sep. 2022). *Simscape Electrical User's Guide (Specialized Power Systems), Release 2022b*. [Online]. Available: https: www.mathworks.com/

[3] J. Mahseredjian, S. Dennetière, L. Dubé, B. Khodabakhchian, and L. Gérin-Lajoie, "On a new approach for the simulation of transients in power systems," *Electric Power Syst. Res.*, vol. 77, no. 11, pp. 1514–1520, Sep. 2007, doi: 10.1016/j.epsr.2006.08.027.

[4] H. W. Dommel, "Section 2-2-2 damping of numerical oscillations with parallel resistance," in *EMTP Theory Book, Microtran Power System Analysis Corporation*, Vancouver, BC, Canada, 1992.

[5] (Feb. 18, 2021). *Modelica—A Unified Object-Oriented Language for Systems Modeling, Language Specification, Version 3.5*. [Online]. Available: https://modelica.org/

[6] A. Junghanns, C. Gomes, C. Schulze, K. Schuch, R. Pierre, M. Blaesken, and M. Najafi, "The functional mock-up interface 3.0-new features enabling new applications," in *Proc. 14th Modelica Conf.*, vol. 202, Sep. 2021, pp. 17–26, doi: 10.3384/ecp2118117.

[7] *Functional Mock-Up Interface*. [Online]. Available: https://fmi-standard.org/

[8] C. C. Pantelides, "The consistent initialization of differential-algebraic systems," *SIAM J. Sci. Stat. Comput.*, vol. 9, no. 2, pp. 213–231, Mar. 1988, doi: 10.1137/0909014.

[9] A. C. Hindmarsh, "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," *ACM Trans. Math. Soft.*, vol. 31, no. 3, pp. 363–396, 2005, doi: 10.1145/1089014.1089020.

[10] P. Fritzson, A. Pop, K. Abdelhak, A. Asghar, B. Bachmann, W. Braun, and P. Ostlund, "The OpenModelica integrated environment for modeling, simulation, and model-based development," *Model., Identificat. Control: A Norwegian Res. Bull.*, vol. 41, no. 4, pp. 241–295, 2020, doi: 10.4173/mic.2020.4.1.

[11] *Dymola: Dynamic Modeling Laboratory*. [Online]. Available: https://www.3ds.com/products/catia/dymola

[12] L. Vanfretti, T. Rabuzin, M. Baudette, and M. Murad, "ITesla power systems library (iPSL): A modelica library for phasor time-domain simulations," *SoftwareX*, vol. 5, pp. 84–88, Jun. 2016, doi: 10.1016/j.softx.2016.05.001.

[13] A. Guironnet, M. Saugier, S. Petitrenaud, F. Xavier, and P. Panciatici, "Towards an open-source solution using modelica for time-domain simulation of power systems," in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. Eur. (ISGT-Europe)*, Sarajevo, Bosnia and Herzegovina, Oct. 2018, pp. 1–6, doi: 10.1109/ISGTEUROPE.2018.8571872.

[14] F. Fachini, M. de Castro, T. Bogodorova, and L. Vanfretti, "OpenIMDML: Open instance multi-domain motor library utilizing the modelica modeling language," *SoftwareX*, vol. 24, Dec. 2023, Art. no. 101591, doi: 10.1016/j.softx.2023.101591.

[15] (2024). *Modelica Standard Library (MSL) Verision 4.1*. [Online]. Available: https://github.com/modelica/ModelicaStandardLibrary

[16] A. Masoom, J. Mahseredjian, T. Ould-Bachir, and A. Guironnet, "MSEMT: An advanced modelica library for power system electromagnetic transient studies," *IEEE Trans. Power Del.*, vol. 37, no. 4, pp. 2453–2463, Aug. 2022, doi: 10.1109/TPWRD.2021.3111127.

[17] A. Masoom, T. Ould-Bachir, J. Mahseredjian, A. Guironnet, and N. Ding, "Simulation of electromagnetic transients with modelica, accuracy and performance assessment for transmission line models," *Electric Power Syst. Res.*, vol. 189, Dec. 2020, Art. no. 106799, doi: 10.1016/j.epsr.2020.106799.

[18] A Masoom, J Gholinezhad, T. Ould-Bachir, J. Mahseredjian, "Electromagnetic transient modeling of power electronics in modelica, accuracy and performance assessment," presented at the 14th Int. Conf. Electrimacs, Nancy, France, 2021, pp. 275–288, doi: 10.1007/978-3-031-24837-5_21.

[19] A. Masoom, T. Ould-Bachir, J. Mahseredjian, and A. Guironnet, "Acceleration of electromagnetic transient simulations in modelica using spatial data locality," *Electric Power Syst. Res.*, vol. 211, Oct. 2022, Art. no. 108577, doi: 10.1016/j.epsr.2022.108577.

[20] A. Masoom, A. Guironnet, A. A. Zeghaida, T. Ould-Bachir, and J. Mahseredjian, "Modelica-based simulation of electromagnetic transients using dynaωo: Current status and perspectives," *Electric Power Syst. Res.*, vol. 197, Aug. 2021, Art. no. 107340, doi: 10.1016/j.epsr.2021.107340.

[21] P. C. Krause, O. Wasynczuk, S D Sudhoff, and S. D. Pekarek, *Analysis of Electric Machinery and Drive Systems*. Hoboken, NJ, USA: Wiley, 2013.

[22] G. J. Rogers and D. Shirmohammadi, "Induction machine modelling for electromagnetic transient program," *IEEE Trans. Energy Convers.*, vol. EC-2, no. 4, pp. 622–628, Dec. 1987, doi: 10.1109/TEC.1987.4765901.

[23] J. C. Das, *Transients in Electrical Systems*. New York, NY, USA: McGraw-Hill, 2010.

ALIREZA MASOOM (Member, IEEE) received the M.A.Sc. and Ph.D. degrees in electrical engineering from Polytechnique Montréal, Montreal, QC, Canada, in 2017 and 2021, respectively. From 2003 to 2015, he was with industrial companies holding various positions in electrical protection, substation automation, and SCADA projects. From 2021 to 2023, he was a Research Assistant with Polytechnique Montréal. In 2023, he joined the Hydro-Québec Research Institute (IREQ) working on the development of real-time and offline electromagnetic transients' simulation tools. He is currently a Registered Engineer in the Province of Quebec. His research interests include electromagnetic transients modeling, computations, and simulation with high-level languages.

JEAN MAHSEREDJIAN (Life Fellow, IEEE) received the M.A.Sc. and Ph.D. degrees in electrical engineering from Polytechnique Montréal, Montreal, QC, Canada, in1985 and 1991, respectively. From 1987 to 2004, he was with IREQ (Hydro Quebec), Quebec, Canada, working on the research and development activities related to the simulation and analysis of electromagnetic transients. In 2004, he joined the Faculty of Electrical Engineering, Polytechnique Montréal. He is the Creator and Lead Developer of EMTP[1] Software.

[1] Registered trademark.