

# MATHEMATICAL SIMULATION OF THE ASYNCHRONOUS ELECTRIC DRIVE OF PERIODIC MOVEMENT

*Vasily O. Nagorniy*<sup>1</sup>, and *Anatoliy V. Aristov*<sup>1,\*</sup>

<sup>1</sup>Tomsk Polytechnic University, 634050, Tomsk, Russia

**Abstract.** The article deals with the approach to construction of mathematical models of electric drives working in the mode of periodic motion. An example of a mathematical model and simulation results are presented.

## 1 Introduction

The modern requirements for drives of antennas of radar stations [1] or optomechanical scanning devices [2] require profound study of the dynamic processes occurring in the motor drives of periodic motion. Since the full-scale pilot studies increase the financial expenses and periods of development and don't allow to conduct comprehensive researches, the use of mathematical simulation in case of complex researches seems to be the most appropriate [3].

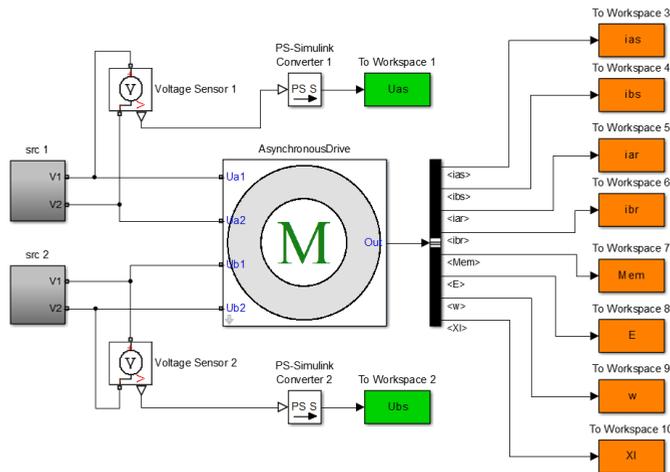
As a system for simulation and research of the electric drive of periodic movement the environment MATLAB/Simulink has been chosen [4, 5].

The article deals with the model of the electric drive of the pulsating movement that is based on a two-phase asynchronous motor (AM), working directly in the mode of periodic motion by the linear phase or balanced amplitude modulation supply voltages to the interruption of one of them in moments of time, when the electromagnetic torque of the AM passes through zero [6]. The complete model of the electric drive consists of electric models of the pulse voltage transformer (PVC) and a two-phase asynchronous motor with a generalized load on the shaft.

The model of the electrical drive of pulsating motion in the mathematical environment, Simulink, implementing phase method when one of the phase windings of the AM from the voltage source is energized through pulse-width voltage converter.

---

\* Corresponding author: [aristovav@tpu.ru](mailto:aristovav@tpu.ru)

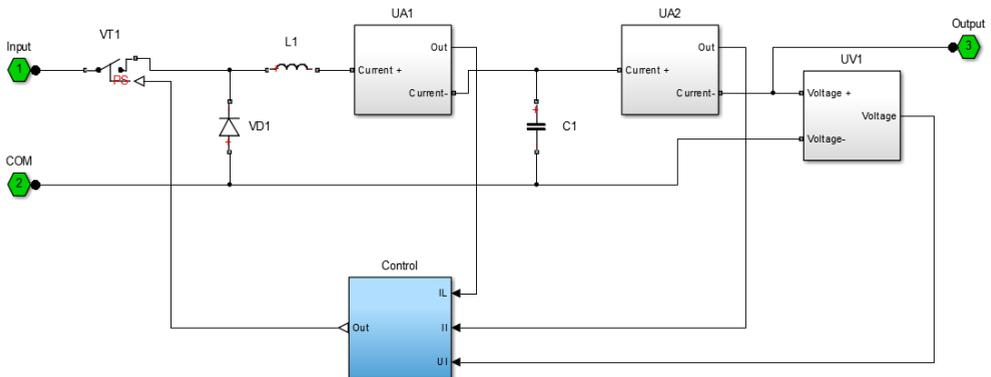


**Fig. 1.** Electric drive model in the Simulink environment.

In Figure. 1 src1, src2 stand for power supplies of phases of the asynchronous motor; AsynchronousDrive stands for the two-phase asynchronous machine with a load on the shaft. The blocks To Workspace are designed to transfer information about the variables Uas, Ubs (indicate phase stator voltage); ias, ibs (are stator phase currents); iar, ibr (are current rotor phase); Mem (stands for the electromagnetic torque of the electric machine); E (is angular acceleration); w (indicates angular velocity); XI (is the movement the motor shaft) in the MATLAB working environment.

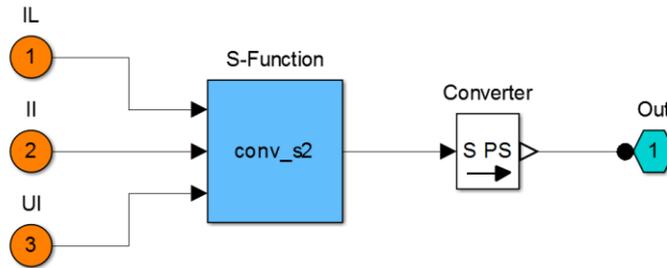
## 2 Simulation of pulsed voltage converter

Model of a pulse voltage converter (Figure. 2) contains power transistor *VT1*, diode *VD1*, filter inductor *L1*, filter capacitor *C1*, current sensors *UA1*, *UA2*, voltage sensor *UV1*, *Control* – a system of the PVC control.



**Fig. 2.** Model of a pulse voltage converter.

While the PVC is being operated, the values of the inductor current, the load current and load voltage enter the system *Control* (Figure. 3), which controls the power transistor *VT1*.



**Fig. 3.** Pulsed voltage converter control system.

In order to integrate the PVC control systems with the mathematical Simulink model, a special approach was selected in which the control system code is executed on a personal computer [7]. The PVC control system is adapted to Level-2 MATLAB S-function. The language used to create a model management system is 'C' [8, 9]. According to model (Figure. 7) S-function is a dynamic link library (dll) [10], exporting the function mexFunction. The mexFunction is executed every time when the interaction of the Simulink with the S-function is required. If the mathematical model is started the S-function launches a thread (by using the CreateThread function) to run the executable code of the PVC control system.

The model calculation runs in the context of the two threads of the operating system. Since the model threads run in parallel, they are forced to keep a record of working time models. To synchronize the time of the threading model the synchronization cycle is introduced. In this case the leading flow is the thread created in the MATLAB environment, and driven thread is that of the control system.

After a specified period of time the model is sure to complete its work by stopping the thread of the control system. In mathematical models of the PVC as well as in the laboratory prototype we use *the same source code* of the control system, *requiring no modifications in its transfer*.

Using the approach described above, the following the PVC control algorithms were tested:

- Proportional-integral-derivative regulation with PWM (pulse-width modulation).
- Controlling the balance of required and the stored energy [11].
- Controlling the balance of required and the stored energy with PWM [12].

### 3 Modeling of the electromechanical drive

While investigating the periodic mode it is convenient to represent the electric machine in the form of a certain idealized generalized model with two couples of co-perpendicular windings: on primary and secondary elements [13], which allows, under certain assumptions, to analyze the greatest number of AM of different types working in a mode of periodic movement.

For this model, there are several options for recording a system of differential equations describing its electrical state. The simplest analysis of equations written in the coordinate axes of  $\alpha, \beta$  or  $d, q$ , rigidly connected to the stator and the rotor, respectively. For example, the coercion to the axes  $\alpha, \beta$  allows to simplify the mathematical description of the process by eliminating periodic coefficients in the equations, describing the state of the electrical windings in the stator AM.

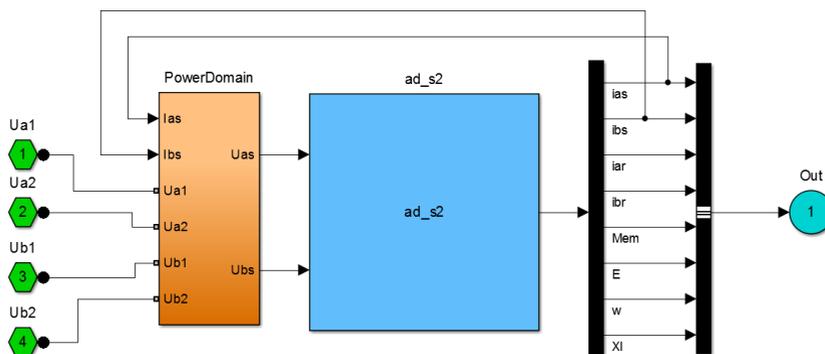
Under conventional assumptions (the magnetic circuit of the machine is unsaturated, the magnetic flux is sinusoidally distributed in space, eddy currents and edge effects are

negligible) [14] the generalized model of the motor, operating in a pulsed mode of motion in the coordinate system  $\alpha, \beta$  is described by the following system of the differential equations:

$$\left. \begin{aligned}
 U_{\alpha s}(t) &= i_{\alpha s}R_{\alpha s} + L_{\alpha s} \frac{di_{\alpha s}}{dt} + M_{\alpha} \frac{di_{\alpha r}}{dt}; \\
 U_{\beta s}(t) &= i_{\beta s}R_{\beta s} + L_{\beta s} \frac{di_{\beta s}}{dt} + M_{\beta} \frac{di_{\beta r}}{dt}; \\
 0 &= i_{\alpha r}R_r + L_r \frac{di_{\alpha r}}{dt} + M_{\alpha} \frac{di_{\alpha s}}{dt} - p\omega(M_{\beta}i_{\beta s} + L_r i_{\beta r}); \\
 0 &= i_{\beta r}R_r + L_r \frac{di_{\beta r}}{dt} + M_{\beta} \frac{di_{\beta s}}{dt} + p\omega(M_{\alpha}i_{\alpha s} + L_r i_{\alpha r}); \\
 M_{em} &= k_d(M_{\alpha}i_{\alpha s}i_{\beta r} - M_{\beta}i_{\beta s}i_{\alpha r}); \\
 L_m \frac{d^2\chi}{dt^2} + R_m \frac{d\chi}{dt} + C_m\chi + M_l &= M_{em},
 \end{aligned} \right\} \tag{1}$$

where  $U_{\alpha s}(t), U_{\beta s}(t), i_{\alpha s}, i_{\beta s}, i_{\alpha r}, i_{\beta r}$  are voltage and current of the phase windings;  $R_{\alpha s}, R_{\beta s}, R_r, L_{\alpha s}, L_{\beta s}, L_r$  are active resistance and total inductance of the phase windings;  $M_{\alpha}, M_{\beta}$  are mutual inductances between the stator and rotor windings on coordinate axes  $\alpha, \beta$ ;  $\omega$  - the rate of change of the generalized coordinates of the movable element;  $M_{em}, M_l$  are generalized electromagnetic force and the load;  $p$  is the number of pairs of poles of the electric machine;  $k_d$  is a generalized power factor;  $L_m$  is an inertia load factor;  $R_m$  is a damping load factor;  $C_m$  is a positional load factor;  $\chi$  is the relocation of the output element of the drive.

For angular movements  $k_d = 1$ , and for the rectilinear  $k_d = \pi^2 / \tau^2$ ; where  $\tau$  is a pole pitch. Now we arrange the system of equations (1) as a Level-2 MATLAB S-Function. Let us examine the internal structure of the mask *AsynchronousDrive* (Figure. 4).



**Fig. 4.** The internal structure *AsynchronousDrive* mask.

The *PowerDomain* subsystem converts the input electrical quantities in dimensionless quantities MATLAB/Simulink. Block *ad\_s2* – block MATLAB S-Function, described in the MATLAB language.

While in operation Simulink environment solver periodically requests the current values of the derivatives of internal states ( $i_{\alpha s}, i_{\beta s}, i_{\alpha r}, i_{\beta r}, \omega, \chi$ ) of S-function, which are calculated

in accordance with **Ошибка! Источник ссылки не найден.**, and updates the state values selected according to the solver.

Such a construction of a mathematical model allows using of different solvers for the same the mathematical description of the model, and changing them if necessary. Further for mathematical simulation the solver *ode45 (Dormand-Prince)* [15, 16] was used.

## Conclusion

The implementation of the proposed approach allows using the algorithm worked out on the mathematical model of a PVC in the control system without additional changes.

Running the algorithm on a personal computer simplifies the process of debugging and enables to use the powerful visualization tools of the various CAD systems.

The application of Simulink environment allows using a visual, graphical representation of a mathematical model; it is easy to alter both the model itself and its parameters.

The construction of mathematical models in MATLAB/Simulink enables to use a variety of solvers for one and the same mathematical description of the model, and change them if necessary.

The power of one of the windings of AM with the DC voltage eliminates the high-frequency pulsation electromagnetic torque that inevitably arises when balanced amplitude or phase modulation of supply voltages are being used and thereby provides a higher energy performance of the drive.

## References

1. V.O. Nagorniy, A.V. Aristov, Herald SibSAU, **16**, 97 (2015)
2. A.V. Aristov, V.O. Nagorniy, Reports of the Tomsk state University of control systems and electronics, **2**, 60 (2013)
3. B.G. Lyubarskiy, E.S. Ryabov, *MATLAB : Materials V Intern. scientific. conf.*, 404 (2011)
4. Simulink – Simulation and Model-Based Design. URL: <http://www.mathworks.com/products/simulink/>
5. V.P. Dyaconov, *Simulink 4. Special handbook* (Piter, St. Petersburg, 2001) [in Russian]
6. A.V. Aristov, *Energy*, **313**, 107 (2008)
7. V.O. Nagorniy, A.V. Aristov, A.M. Gavrillov, Electromechanical energy converters: materials VII Intern. scientific and engineering. Conf., 201 (2015).
8. B. Kernighan, D. Ritchie, *The C Programming Language* (Williams, 2007)
9. ISO / IEC 9899: 2011 – Information technology – Programming languages
10. What is a DLL? URL: <https://support.microsoft.com/ru-ru/kb/815065>
11. A.F. Lekarev, V.O. Nagorniy, A.V. Aristov, VI Intern. scientific and engineering. conf., 192 (2013)
12. Yu.M. Kazantsev, A.F. Lekarev, Devices and systems. Management, control, diagnostics, **5**, 17 (2009)
13. I.P. Kopylov, *Electromechanical energy converters* (Energy, St. Petersburg, 1973) [in Russian]
14. A.K. Gromov, A.G. Ershov, A.V. Likhacheva, Electromechanical energy conversion: tutorial. Ivanovo, 2008.
15. J.R. Dormand, Numerical Methods for Differential Equations: A Computational Approach. Boca Raton: CRC Press, 1996.
16. A.E. Novikov, Development of algorithms of variable structure for solving stiff problems: Dis. Cand. Physics and Mathematics. Sciences. Krasnoyarsk, 2014.