

## SIMULINK AS AN ADVANCED TOOL FOR ANALYSIS OF DYNAMICAL ELECTRICAL SYSTEMS

Stanislaw Osowski

Warsaw University of Technology, Military University of Technology  
sto@iem.pw.edu.pl

**Abstract.** Simulink is a powerful toolbox of Matlab, very useful in analysis of dynamical systems. The paper presents the application of Simulink/Matlab in analysis of the dynamical electrical systems. It will be shown that it is an ideal tool for quantitative and qualitative analysis of electrical and electronic networks. Its usefulness will be shown on the example of modeling the 3-phase induction machine and a power system. The non-classical examples of using Simulink include the analysis of nonlinear system in the state space, determination of the stability of the equilibrium points as well as optimization of the electrical circuit. It should be mentioned that nowadays version of Simulink allows also to model the electrical networks directly on the basis of circuit structure (the specialized library *SimElectronics*) without describing explicitly the circuit by the system of equations.

**Key words:** Dynamical electrical systems, simulation, numerical methods, state space analysis.

### 1. Introduction

Simulink is an extension of Matlab in the form of toolbox used to build the computer models of the dynamical systems by applying the user friendly graphical block diagram notations [2, 3, 4]. Simulink converts this graphical representation into a state space representation consisting of the set of simultaneous first order differential or difference equations. These equations are then solved by using the Matlab integrating functions [7, 9].

Simulink enables to simulate in an easy way both continuous and discrete systems on the basis of either state space or transfer function description of the individual blocks forming the system under analysis [12]. The common approach of using Simulink in solution of any problem of process modeling involves few stages:

a) Description of the modeled process by the set of differential equations or in the form of transfer function description (s-domain for continuous and z-domain for discrete systems);

b) Building the block diagrams representing this mathematical description of the system in the Simulink space, including the graphical presentation of results.

c) Solution of the problem using the solver built in the Simulink platform.

Simulink contains many blocks representing different elements collected in specialized libraries used in practice of simulation of many types of systems, including the dynamical systems. These blocks enable to build different, even the most complex nonlinear models of the dynamical systems. Moreover, some blocks may be combined into subsystems and stored as the separate subsystems for future use (so called masked blocks). They provide the capability within Simulink similar to subprograms in traditional programming languages making subsystems easier to use and understanding.

Moreover Simulink allows to incorporate the existing Matlab codes into a Simulink model in the form of so called S-functions [3,6]. They may be used in situation where it is easier to describe a subsystem algorithmically than in a block diagram notation. It is possible to write S-functions either as Matlab function M-files or by using the C programming language via the Matlab MEX-file mechanism.

The paper reviews the Simulink application in modeling dynamical systems. The considerations will be illustrated by the examples of simulating the 3-phase induction machine and the macro model of the cooperation of two power systems in stabilization of the frequency. Two additional non-typical applications will be concerned with the cooperation of Simulink and Matlab to determine the equilibrium points of the dynamical systems as well solving some optimization problem in electrical circuit.

### 2. Building the Simulink model

The typical application of Simulink in modeling a dynamical system requires to describe the system using the ordinary differential equations. The best way of such description is the state space model

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{f}(\mathbf{x}, t) \\ \mathbf{y} &= \mathbf{g}(\mathbf{x}, t) \end{aligned} \quad (1)$$

where  $\mathbf{x}$  is the  $n$ -dimensional state vector and  $\mathbf{f}(\mathbf{x}, t)$  describes the nonlinear process under modeling (also  $n$ -dimensional). The second nonlinear equation represents

the output function of the system. Fig. 1 presents the general block diagram of solution of the equations described by (1). The function  $f(x,t)$  represents the right side of the differential equation and depends on the state space vector  $x$  and the time  $t$ . After solving the equation (1) using the integrators we get values of the state variables (vector  $x$ ) which are used to build both  $f(x,t)$  and output function  $g(x,t)$ . In building these functions we can use all blocks available in Simulink.

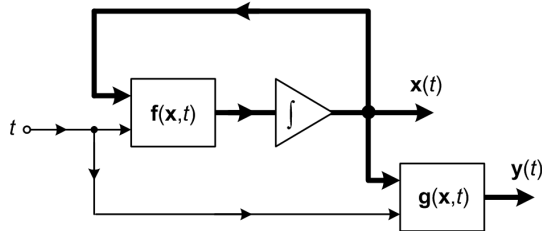


Fig. 1. The general scheme of solution of differential equation (1)

There is variety of libraries of blocks grouped together according to the subject they represent. To the most important belong the libraries representing the continuous, discontinuous and discrete elements, blocks of mathematical operations, ports and subsystems, signal routing elements, set of different sources and sinks as well as user defined functions. All of them are user friendly and easy to use in building the required block diagrams, superficially resembling the computer program flow chart, although conceptually quite different. In building the Simulink model we drag the blocks from the appropriate libraries to the working space and then connect them by using the mouse.

After building the model of the simulated system we can set the values of the parameters of the blocks and then we are ready to solve them by using the built in solver. The solution of the differential equations represented by the Simulink model may be achieved by different integration procedures built into Matlab by applying either variable or constant step. In the case of continuous systems we use mainly variable step methods. Actually there are following integration procedures available for application [9,10,12]:

ode45 – variable step Runge-Kutta method of 4th and 5th order in Dormand-Prince implementation

ode23 - variable step Runge-Kutta method of 2nd and 3rd order in Bogacki-Shampine implementation

ode113 – the multistep variable order predictor-corrector method of Adams-Bashforth (prediction) and Adams-Moulton (correction)

ode23t – trapezoidal method of integration

ode23s – modified Rosenbrock method suited for stiff equations

ode23tb – the cooperation of the trapezoidal rule and Gear method of integration suited for stiff equations

ode15s – Klopfenstein method for solution of stiff equations

This suit for solving the set of differential equations is extremely efficient and generates stable solution of different complexity equations, including the stiff ones.

### 3. Example of application of Simulink in modeling the dynamical processes

In this section we will present two examples of using Simulink in modeling the dynamical system [10]. The first one will be concerned with 3-phase induction machine (the nonlinear model), where the starting point is the state space description of the machine. The second one is the linearized model of two power systems cooperating in stabilization of the frequency in the power network.

#### 3.1 Dynamical model of the induction machine

In building this model we have assumed some simplifying assumptions: symmetry of the 3-phase windings of stator and rotor, sinusoidal change of flux in the air gap of the machine, neglecting the nonlinearity effects of magnetic materials, assuming the constant values of self- and mutual inductances of the machine, neglecting the eddy currents and anisotropy of the magnetic materials.

If we denote respectively by  $u_{si}$ ,  $i_{si}$ ,  $\psi_{si}$  ( $i = 1, 2, 3$ ) the voltage, current and flux linkage of the 3-phase windings of the stator and by  $R_s$  the stator resistance we get 3 differential equations describing the stator windings

$$u_{si} = \frac{d\psi_{si}}{dt} + R_s i_{si} \quad (2)$$

for  $i=1, 2, 3$ . Identical equations are valid for rotor windings, for which the index  $s$  is replaced by  $r$

$$u_{ri} = \frac{d\psi_{ri}}{dt} + R_r i_{ri} \quad (3)$$

We get full system of description of the machine by writing the mechanical equation of motion

$$J \frac{d\Omega_m}{dt} = M_e(t) - M_m(t) \quad (4)$$

where  $J$  is the moment of inertia,  $M_m$  – the mechanical torque representing the load,  $M_e$  – the electromagnetic torque generated by the rotating machine and  $\Omega_m$  – the angular mechanical speed of the rotor related to the electrical angular speed  $\Omega$  in the form

$$\Omega_m = \frac{\Omega}{p}, \text{ where } p \text{ denotes the number of pole pairs of}$$

the machine. By introducing the complex space vectors of current, voltage and flux perpendicular to the engine

shaft (defined for the stator current  $\mathbf{i}_s = \frac{2}{3}(\mathbf{i}_{s1} + a\mathbf{i}_{s2} + a^2\mathbf{i}_{s3})$  and in the same fashion for other variables) we can simplify 6 scalar equations defined by (2) and (3) into two vector equations

$$\begin{aligned}\mathbf{u}_s &= R_s \mathbf{i}_s + \frac{d\boldsymbol{\Psi}_s}{dt} \\ \mathbf{u}_r &= R_r \mathbf{i}_r + \frac{d\boldsymbol{\Psi}_r}{dt}\end{aligned}\quad (5)$$

To define common platform for stator and rotor we introduce the new coordinate system rotating with the synchronous speed and denote it with the upper index  $k$ . They are given in the form  $\mathbf{i}_s = \mathbf{i}_s^{(k)} e^{j\theta_k}$ ,  $\mathbf{i}_r = \mathbf{i}_r^{(k)} e^{j(\theta_k - \theta)}$ ,  $\mathbf{u}_s = \mathbf{u}_s^{(k)} e^{j\theta_k}$ ,  $\mathbf{u}_r = \mathbf{u}_r^{(k)} e^{j(\theta_k - \theta)}$ ,  $\boldsymbol{\Psi}_s = \boldsymbol{\Psi}_s^{(k)} e^{j\theta_k}$ ,  $\boldsymbol{\Psi}_r = \boldsymbol{\Psi}_r^{(k)} e^{j(\theta_k - \theta)}$  where  $j = \sqrt{-1}$  is the unity complex operator,  $\theta$  represents the actual shaft angle and  $\theta_k$  the synchronous shaft angle. Taking into account that  $\Omega = \frac{d\theta}{dt}$ , and the synchronous speed  $\Omega_s = \omega_s = \frac{d\theta_k}{dt}$  the equations (5) may be transformed to the vector form

$$\frac{d\boldsymbol{\Psi}_s^{(k)}}{dt} = -j\Omega_s \boldsymbol{\Psi}_s^{(k)} - R_s \mathbf{i}_s^{(k)} + \mathbf{u}_s^{(k)} \quad (6)$$

$$\frac{d\boldsymbol{\Psi}_r^{(k)}}{dt} = -j(\Omega_s - \Omega) \boldsymbol{\Psi}_r^{(k)} - R_r \mathbf{i}_r^{(k)} + \mathbf{u}_r^{(k)}$$

or scalar form

$$\frac{d(\Psi_{s\alpha}^{(k)} + j\Psi_{s\beta}^{(k)})}{dt} = -j\Omega_s (\Psi_{s\alpha}^{(k)} + j\Psi_{s\beta}^{(k)}) - R_s (\mathbf{i}_{s\alpha}^{(k)} + j\mathbf{i}_{s\beta}^{(k)}) + (\mathbf{u}_{s\alpha}^{(k)} + j\mathbf{i}_{s\beta}^{(k)}) \quad (7)$$

$$\frac{d(\Psi_{r\alpha}^{(k)} + j\Psi_{r\beta}^{(k)})}{dt} = -j(\Omega_s - \Omega) (\Psi_{r\alpha}^{(k)} + j\Psi_{r\beta}^{(k)}) - R_r (\mathbf{i}_{r\alpha}^{(k)} + j\mathbf{i}_{r\beta}^{(k)}) + (\mathbf{u}_{r\alpha}^{(k)} + j\mathbf{i}_{r\beta}^{(k)})$$

The lower index  $\alpha$  in the last equations represents the real part and  $\beta$  does the imaginary part of the appropriate variable. To get final set of equations we have to consider the linear relations between current and flux of the stator and rotor, which will be presented in a matrix form

$$\begin{bmatrix} \mathbf{i}_{s\alpha}^{(k)} + j\mathbf{i}_{s\beta}^{(k)} \\ \mathbf{i}_{r\alpha}^{(k)} + j\mathbf{i}_{r\beta}^{(k)} \end{bmatrix} = \frac{1}{L_s L_r - L_m^2} \begin{bmatrix} L_r & -L_m \\ -L_m & L_s \end{bmatrix} \begin{bmatrix} \Psi_{s\alpha}^{(k)} + j\Psi_{s\beta}^{(k)} \\ \Psi_{r\alpha}^{(k)} + j\Psi_{r\beta}^{(k)} \end{bmatrix} \quad (8)$$

where  $L_s$  is the inductance of one phase of the stator,  $L_r$  – the inductance of the rotor phase transformed to the stator side and  $L_m$  – the magnetizing inductance. The

electromagnetic torque developed by the machine is then described in the following form

$$M_e = \frac{3pL_m}{2(L_s L_r - L_m^2)} \left[ \Psi_{s\beta}^{(k)} \Psi_{r\alpha}^{(k)} - \Psi_{s\alpha}^{(k)} \Psi_{r\beta}^{(k)} \right] \quad (9)$$

Ordering the equations (7), (8) and (9) we get the final dynamical description of the machine in the following normal state space form [10].

$$\begin{aligned}\frac{d\Psi_{s\alpha}^{(k)}}{dt} &= -\alpha \Psi_{s\alpha}^{(k)} + \alpha K_r \Psi_{r\alpha}^{(k)} + \Omega_s \Psi_{s\beta}^{(k)} + \mathbf{u}_{s\alpha}^{(k)} \\ \frac{d\Psi_{s\beta}^{(k)}}{dt} &= -\alpha \Psi_{s\beta}^{(k)} + \alpha K_r \Psi_{r\beta}^{(k)} - \Omega_s \Psi_{s\alpha}^{(k)} + \mathbf{u}_{s\beta}^{(k)} \\ \frac{d\Psi_{r\alpha}^{(k)}}{dt} &= \beta K_s \Psi_{s\alpha}^{(k)} - \beta \Psi_{r\alpha}^{(k)} + (\Omega_s - \Omega) \Psi_{r\beta}^{(k)} + \mathbf{u}_{r\alpha}^{(k)} \\ \frac{d\Psi_{r\beta}^{(k)}}{dt} &= \beta K_s \Psi_{s\beta}^{(k)} - \beta \Psi_{r\beta}^{(k)} - (\Omega_s - \Omega) \Psi_{r\alpha}^{(k)} + \mathbf{u}_{r\beta}^{(k)} \\ \frac{d\Omega}{dt} &= \frac{3p^2 K_r}{2\sigma L_s J} \left[ \Psi_{s\beta}^{(k)} \Psi_{r\alpha}^{(k)} - \Psi_{s\alpha}^{(k)} \Psi_{r\beta}^{(k)} \right] - \frac{P}{J} M_m\end{aligned}\quad (10)$$

where we have introduced:  $\sigma = 1 - \frac{L_m^2}{L_s L_r}$ ,

$$K_s = \frac{L_m}{L_s}, \quad K_r = \frac{L_m}{L_r}, \quad \alpha = \frac{R_s}{\sigma L_s}, \quad \beta = \frac{R_r}{\sigma L_r}.$$

Associating the coordinate system with the synchronous speed we may assume  $\mathbf{u}_{s\alpha}^{(k)} = U_m$  and  $\mathbf{u}_{s\beta}^{(k)} = 0$ . Additionally we assume that the supply is attached only to the stator, and it means  $\mathbf{u}_{r\alpha}^{(k)} = \mathbf{u}_{r\beta}^{(k)} = 0$ . Fig. 2 presents the Simulink model representing this system of differential equations and the blocks generating the electromagnetic torque  $M_e$ .

In practical experiments we have assumed the machine of the following parameters: the real power  $P=1700\text{W}$ , number of pole pairs  $p=3$ , three-phase supply voltage of  $V_{ph}=220\text{V}$  and  $f=50\text{Hz}$ . The internal parameters of the machine were as follows:  $R_s=3.57\Omega$ ,  $R_r=3.68\Omega$ ,  $L_s=0.284\text{H}$ ,  $L_r=0.298\text{H}$ ,  $L_m=0.262\text{H}$ , moment of inertia  $J=0.31\text{kg}\cdot\text{m}^2$ . It means that  $\sigma=0.1889$ ,  $K_s=0.9225$ ,  $K_r=0.8792$ ,  $\alpha=66.5454$ ,  $\beta=65.3732$ ,  $u_{s\alpha}^{(k)} = U_m = \sqrt{2} \cdot 220 = 311.3\text{V}$ .

The electromagnetic torque developed by the engine is described then by  $M_e = 73,74(\Psi_{s\beta}^{(k)} \Psi_{r\alpha}^{(k)} - \Psi_{s\alpha}^{(k)} \Psi_{r\beta}^{(k)})$ . The mechanical speed is equal  $\Omega_m=1/3\Omega$ . At  $f=50\text{Hz}$  the synchronous speed of the machine is equal  $\Omega_{ms} = 104.7\text{rad/s}$ , and the nominal torque  $M_n=P/\Omega_{ms}=16.24\text{Nm}$

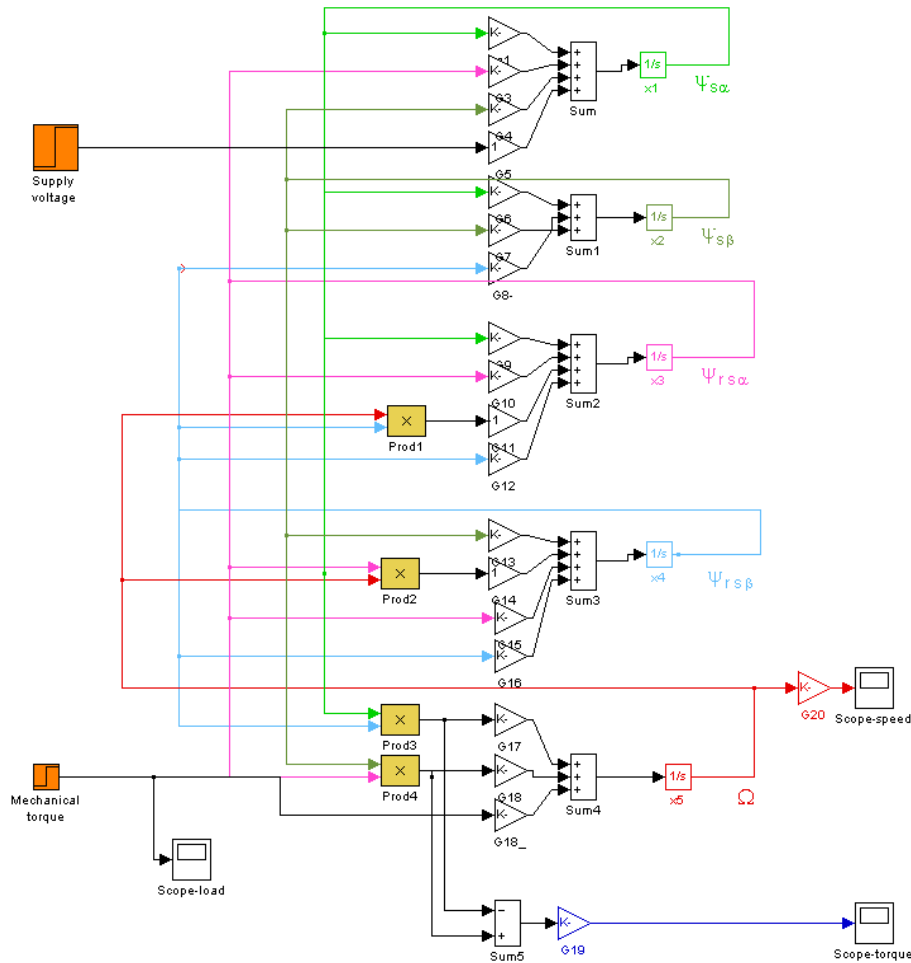


Fig. 2. The Simulink model of the 3-phase induction machine

The above presented model can be the basis for many different numerical experiments investigating its behavior in dynamical states (start, changing the supply and load, stopping, etc.). As an example in Fig. 3 we present the change of the angular speed  $\Omega_m$  and electromagnetic torque  $M_e$  of the machine at different values of the mechanical torque of the load. Two time constants characterizing the electromagnetic and electromechanical processes are visible. At the beginning of the start the electromagnetic processes dominate. Later we observe the steady state of electromagnetic process and only mechanical transient is visible. The total time of transient in this type of machine is strictly dependent on the load. The higher the load the longer is the time of stabilization of the speed.

The next figure (Fig. 4) illustrates the change of the speed and the electromagnetic torque at changing value of the mechanical load. In these experiments we have assumed the following values of  $M_m$ : zero (at the start of machine) and then every 2 seconds change to 15Nm, 25Nm and 35Nm. The value  $M_m=35$  Nm exceeded the maximum torque of the machine and the motion of the

machine was following the dynamics governed by the negative value of the equivalent driving torque  $M_e - M_m$ .

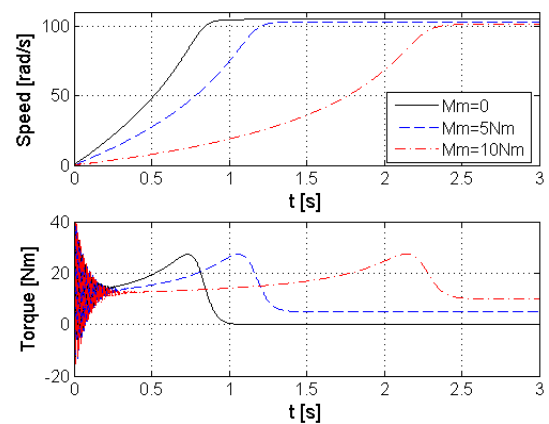


Fig. 3. The transient of the speed  $\Omega_m$  and electromagnetic torque  $M_e$  of the machine at different values of the mechanical torque of the load

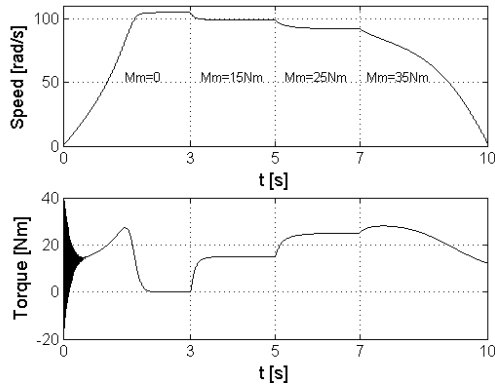


Fig. 4. The change of the speed  $\Omega_m$  and the electromagnetic torque  $M_e$  at changing values of  $M_m$

### 3.2 Dynamical model of cooperation of two power systems at frequency stabilization.

The next example will present the linearized system described by the transfer function in s-domain. Fig. 5 presents the general functional diagram of the power generating unit [1, 5, 10] with the frequency stabilization system in the form of PI feedback regulator. The whole system contains the generator, turbine, hydraulic cylinder (amplifier) controlling the flow of the gas in turbine and the regulation system PI of the frequency composed of proportional and integrating regulator. The proportional block of this regulator has the gain  $k=1/R$ , where  $R$  the construction dependent parameter. The integrating part of the regulator PI adds up all differences between the actual and reference frequency and is described by the transfer function  $-K_i/s$  of the gain  $K_i$  adjusted by the user.

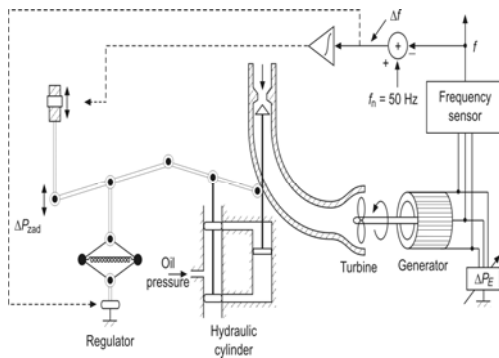


Fig. 5. The general functional scheme of the power generating unit

To build the simplified model of the whole system the linearized first order models of the nonlinear elements have been applied. In this way the hydraulic cylinder, turbine and generator have been represented by the following first order transfer functions.

Hydraulic cylinder

$$T_H(s) = \frac{K_H}{1 + sT_H} \quad (11)$$

Turbine

$$T_T(s) = \frac{K_T}{1 + sT_T} \quad (12)$$

Generator

$$T_G(s) = \frac{K_G}{1 + sT_G} \quad (13)$$

The linearization was performed at the operating point of these devices. The resulting transfer function incremental model of the whole system is illustrated on Fig. 6. The signals of the models represent the increments of all variables (frequency, power, etc.). This scheme is directly implemented as a Simulink model, using the elements already existing in the library of the program.

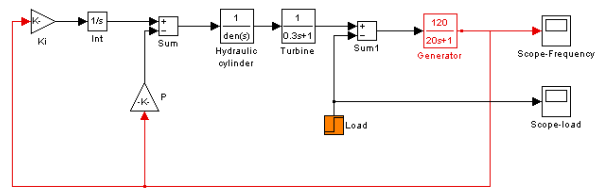


Fig. 6. The linearized transfer function model of the power generating unit of Fig. 5

In practice many generating subsystems may cooperate with each other, supporting one which is in highest degree loaded. In this way the reduction of the temporary change of the local frequency of the most loaded subsystem is possible. The less loaded subsystem has higher local frequency, resulting from the change of the shaft angle of the generator with respect to the more loaded units. The difference of the shaft angles between two units causes the flow of the balancing power between them. The flow of this balancing power  $\Delta P_{12}$  between two systems is described by the relation proportional to the difference between the local frequencies and may be described by

$$\Delta P_{12} = T_0 [\Delta \theta_1(s) - \Delta \theta_2(s)] = \frac{2\pi T_0}{s} [\Delta f_1(s) - \Delta f_2(s)] \quad (14)$$

Fig. 7 presents the Simulink model of two generating units containing the PI feedback regulator, stabilizing the frequency.

In the numerical experiments we have assumed the data corresponding to the power station of the nominal power 1000MW [1, 10]. For this station the typical values of the parameters are as follows:  $K_H=K_T=1$ ,  $K_G=120\text{Hz/MW}$ ,  $T_H=80\text{ms}$ ,  $T_T=300\text{ms}$ ,  $T_G=20\text{s}$ ,  $R=2.4\text{Hz/MW}$ ,  $T_0=0.0707\text{MW/rad}$ ,  $B=0.425\text{MW/Hz}$ . The gain of the integrating block  $K_i$  was adjusted in a way to provide the shortest period of transient of the controlled frequency. At the assumed values of parameters of the system the optimal value of  $K_i$  was equal 0.25.

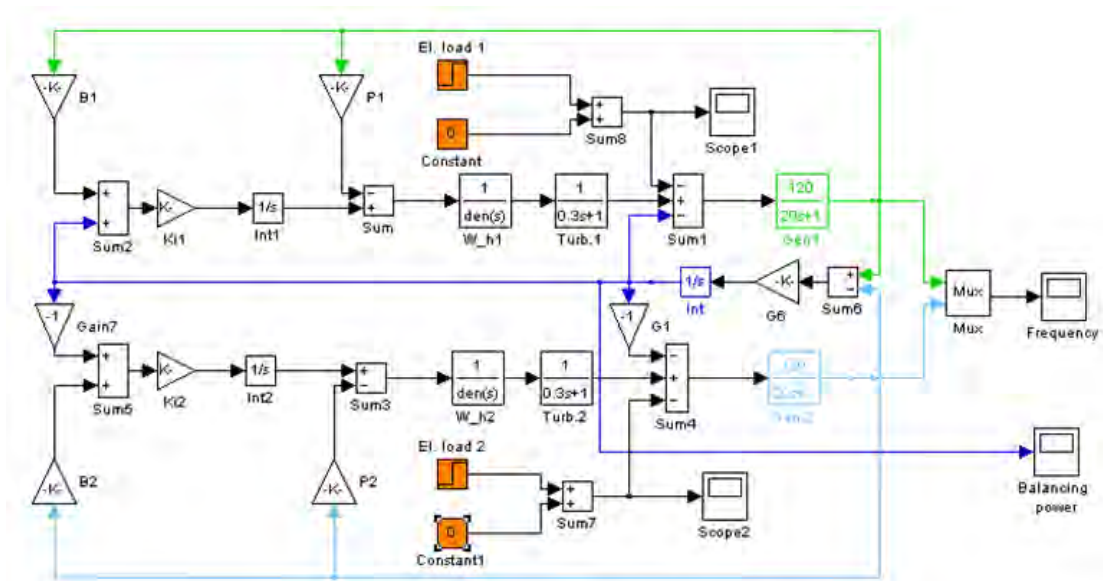


Fig. 7. The Simulink linearized model of two generating units containing the frequency stabilizing PI feedback

Fig. 8 presents the exemplary transient signals representing the dynamical change of the local frequencies of two cooperating power stations and the flow of the balancing power  $\Delta P_{12}$  at the abrupt change of the load of the first station  $\Delta P_{el}=0.2$  (in relative terms) and at unchanged load of the second one. The balancing power flowing from the second to the first subsystem is expressed in normalized form.

Observe that the second subsystem for which the load was unchanged, has automatically loaded itself, supporting the first one in its need for additional power. Thanks to this the change of the local frequency of the first subsystem has been significantly reduced. Thanks to the integrating regulator the changes of both local frequencies disappear after short period of time and both systems adapt themselves automatically to the new operating conditions of loading. As a result the cooperation of two subsystems allows to reduce the magnitude of the frequency change and to accelerate the return to the nominal operating frequencies of both systems.

Fig. 9 presents the variation of the local frequencies and the flow of the balancing power at the cumulative simultaneous changes of the load of both systems. Fig. 9a illustrates the applied changes of the load, Fig. 9b – the local frequencies and Fig. 9c – the flow of the balancing power.

The presented above solutions show the capability of both subsystems to support each other in stabilizing the frequency at sudden change of the load, that may happen in practice. Observe that after short period of time the

flow of the balancing power is automatically stopped and does not lead to permanent overloading of the smaller subsystem.

Summarizing we may conclude that application of Simulink is very efficient in modeling the behavior of the power system at different operating conditions. It enables to investigate different dynamical states of the system composed of arbitrary number of the cooperating subsystems, delivering the results in a very short time. It is an ideal tool for learning the stability of the system at changing load.

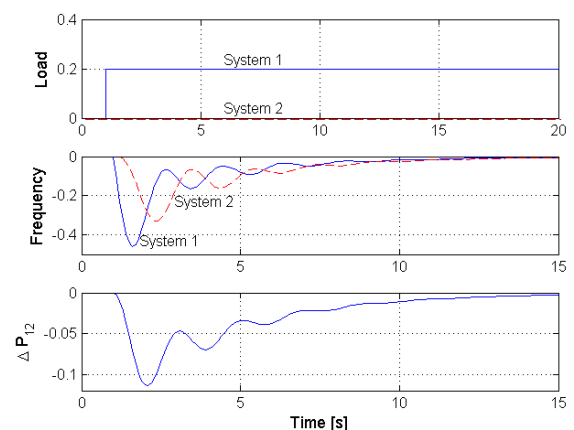


Fig. 8. The dynamic behavior of two cooperating systems (a) the actual load of both systems, b) the change of local frequencies of both systems, c) the flow of balancing power at the abrupt change of the load of the first system

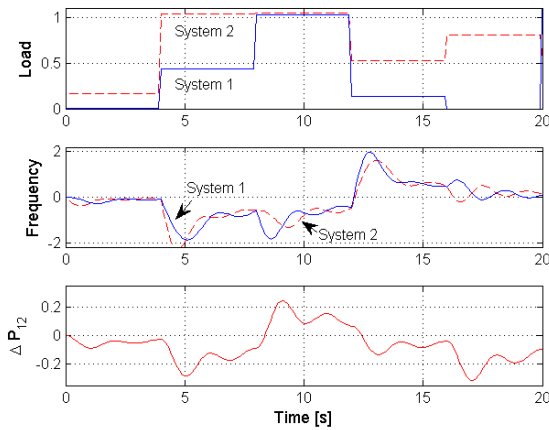


Fig. 9. The change of the temporary signals of two cooperating power stations at cumulative simultaneous changes of the load of both stations: a) the change of the load, b) local frequencies, b) balancing power

#### 4. Determination of equilibrium points in dynamical system

At this point we will present the non-typical application of Simulink to investigate the equilibrium points of the nonlinear dynamical systems [2]. It is done through the application of the Matlab function *trim* determining the equilibrium closest to the initial point and the function *linmod* used for linearization of the nonlinear system (the matrices **A**, **B**, **C**, **D** at the operating point of the nonlinear system). Cooperation of Simulink with the Matlab platform allows to investigate the local stability of the equilibrium points on the basis of the location of the eigenvalues determined using Matlab *eig* function. The whole procedure may be summarized in few stages [11].

The first point in this application is the creation of the nonlinear system model of Simulink and storing it as the *mdl* file (for example as *non\_sys.mdl*).

Next call the *trim* function giving the name of file (for example *non\_sys.mdl*) and initial conditions  $\mathbf{x}_{start}$  of the vector  $\mathbf{x}$  from which the search of the equilibrium will start. Its calling may take the form

$$\mathbf{x} = \text{trim}(\text{'non\_sys'}, \mathbf{x}_{start})$$

The resulting vector  $\mathbf{x}$  returns the determined equilibrium point. At this point we apply the *linmod* function linearizing the nonlinear state space description of the system by using the command

$$[A, B, C, D] = \text{linmod}(\text{'non\_sys'}, \mathbf{x})$$

The local stability of the system is then defined on the basis of the eigenvalues of the matrix **A**: position of all eigenvalues in the left-hand side of the complex plane means stability, one or more eigenvalues in the right-hand side – instability, position of eigenvalues on the imaginary axis means that the linear approximation of the system is not appropriate for stability assessment of the equilibrium point.

As an example let us consider the nonlinear system described in the following form

$$\begin{aligned} \frac{dx_1}{dt} &= 2x_1^2 + x_2^2 + x_1 + 10 \\ \frac{dx_2}{dt} &= 3x_1 - x_2 \end{aligned} \quad (15)$$

The Simulink model of it is presented in Fig. 10. It is stored in the file named *non\_sys.mdl*.

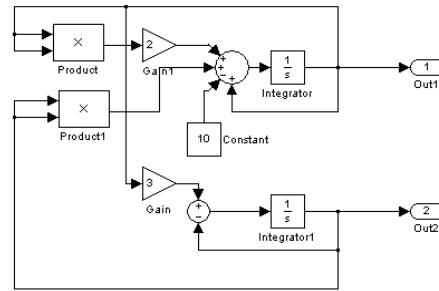


Fig. 10. The Simulink model *non\_sys.mdl* of the nonlinear system described by the equation (12)

To determine the starting positions of the *trim* function we create first the phase portrait of the system using Matlab. The appropriate fragment of the program is given below.

```
% Phase portrait of the nonlinear system
h=0.01;
opt=simset('Solver','ode5','FixedStep',h);
x1=-4:0.25:4;
x2=-4:0.25:4;
[nw,nk]=size(x1);
x1m=zeros(nk,nk);
x2m=x1m;
for nx1=1:nk
for nx2=1:nk
opt=simset(opt,'InitialState',[x1(nx1),x2(nx2)]);
[t,x,y]=sim('non_sys',h,opt);
dx1=x(2,1)-x1(nx1);
dx2=x(2,2)-x2(nx2);
s=sqrt(dx1^2+dx2^2)*10; % scaling of the arrow
if s>1e-10
x1m(nx2,nx1)=dx1/s;
x2m(nx2,nx1)=dx2/s;
end
end
end
quiver(x1,x2,x1m,x2m,0); % drawing the succeeding
positions of the trajectory
axis([-4, 4, -4, 4]);
xlabel('x1')
ylabel('x2')
grid
The phase portrait shown in Fig. 11 indicates two
equilibrium points denoted by the red rectangles. Their
exact positions can be determined using Matlab
command trim
x1=trim('non_sys',[-3, -5])
x2=trim('non_sys',[-3, 4])
```



As a results we get

$$\mathbf{x}_1 = \begin{bmatrix} -1 \\ -3 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 0.91 \\ 2.72 \end{bmatrix}$$

At each point we linearize the system using the *linmod* command of Matlab.

```
[A1, B1, C1, D1]=linmod('non_sys', x1)
[A2, B2, C2, D2]=linmod('non_sys', x2)
```

Applying the *eig* function for both state matrices we get their eigenvalues  $\lambda$

```
eig1=eig(A1)
eig2=eig(A2)
```

As a result we get the eigenvalues of both matrices

For matrix  $\mathbf{A}_1$  (the equilibrium point  $\mathbf{x}_1$ )

$$\lambda_1 = -2 + j4.12, \lambda_2 = -2 - j4.12$$

For matrix  $\mathbf{A}_2$  (the equilibrium point  $\mathbf{x}_2$ )

$$\lambda_1 = 6.75, \lambda_2 = -3.11$$

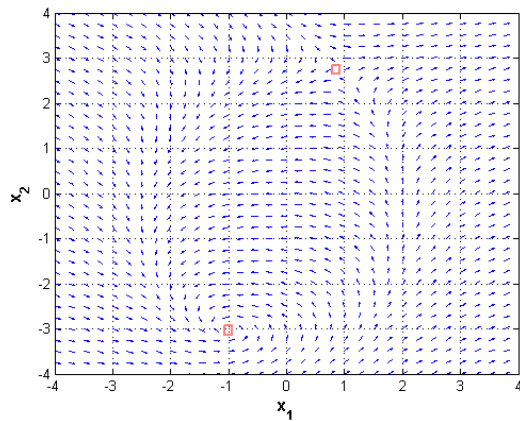


Fig. 11. The phase portrait of the nonlinear system

The first equilibrium point corresponding to the matrix  $\mathbf{A}_1$  is stable (both eigenvalues in left-hand side of the space plane) and the second (the matrix  $\mathbf{A}_2$ ) is unstable (one eigenvalue placed in the right-hand side of the space plane).

### 5. Application of Simulink in optimization problem

The Simulink may be easily adopted for cooperation with the Matlab optimization toolbox [8] in solving different kinds of optimization problems. In such case the analysis of the network is done by the Simulink program and the results of this analysis are delivered through the workspace of Matlab to the optimization functions. We show this utility on the example of adjusting the resistance of the series RLC circuit to provide the shortest possible period of the transient in this circuit at switching on the DC voltage source  $e(t)=E$ . The analysis problem is linear. Assuming state variables as the inductor current  $i$  and the capacitor voltage  $u$  we get the state space description of the problem in the following form.

$$\begin{bmatrix} \frac{di}{dt} \\ \frac{du}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix} \begin{bmatrix} i \\ u \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} e(t) \quad (16)$$

As the output signal in further considerations we take the current  $i(t)$ . It means that the matrix  $\mathbf{C}$  and  $\mathbf{D}$  of the state description is given as  $\mathbf{C}=[1 \ 0]$ ,  $\mathbf{D}=0$ . The Simulink model (*rlc.mdl*) allowing the analysis of transient in this circuit is presented in Fig. 12.

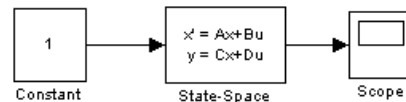


Fig. 12. The Simulink model of the series RLC circuit

Our task is to optimize the value of the resistance  $R$  at constant values of  $L=1\text{H}$ ,  $C=1\text{F}$  and  $E=1$  to get the shortest period of transient current in the circuit. We assume that the current achieves its steady state when its value is continuously below some a priori assumed threshold  $\varepsilon$ . Different values of  $\varepsilon$  will be tried in the experiments.

To solve this problem we have used the function file *rlc\_opt.m* in Matlab defining the objective function under minimization. This function determines the actual time of achieving the steady state with the threshold (tolerance)  $\varepsilon$ . The solution applied in experiments is coded in the Matlab program in the following form.

```
% RLC – Optimization function using Simulink
function f=rlc_opt(R)
t0=0; tf=50; h=0.01;
opts=simset('SRCWorkspace','current');opts=simset(op
ts,'DSTWorkspace','current');
[t,x]=sim('rlc',[t0 h tf],opts);
epsilon=0.0001;
tmx=find(abs(x(:,1))<epsilon);
ntmx=length(tmx);
for i=ntmx:-1:2
if(tmx(i)-tmx(i-1))==1)
p=tmx(i);
else break;
end
end
f=t(p);
```

Start of the optimization procedure is initiated by writing the following command in Matlab workspace

```
[R, tmin]= fminbnd('rlc_opt',0.4, 3);
```

As a result of optimization routine we get the optimal value of the resistance  $R$  and also the minimal time  $t_{min}$  of achieving the steady state in RLC circuit (both returned by the optimization function). It should be observed that the optimal value of resistance  $R$  depends on the assumed threshold  $\varepsilon$ . Table 1 presents the dependence of the optimal value of  $R$  and the minimum



time  $t_{min}$  of achieving the steady state in RLC circuit at  $L=1H$  and  $C=1F$  as a function of the assumed value of  $\varepsilon$ .

Table 1

The dependence of the optimal value of R and  $t_{min}$  on the threshold  $\varepsilon$

| $\varepsilon$ | R [ $\Omega$ ] | $t_{min}$ [s] |
|---------------|----------------|---------------|
| 0.01          | 1.53           | 4.49          |
| 0.005         | 1.62           | 5.13          |
| 0.001         | 1.77           | 6.47          |
| 0.0001        | 1.87           | 8.56          |
| 0.00001       | 1.91           | 10.73         |
| 0.000001      | 1.94           | 12.94         |
| 0.0000001     | 1.97           | 17.45         |

It is seen that when  $\varepsilon$  decreases to zero we approach the ideal theoretical value of resistance equal to the critical value of  $2\Omega$ . Observe that Simulink allows one to adjust the optimal value of the parameters in any dynamical circuit with respect to the defined criterion of optimality. The task of user is only to build the dynamical system model in Simulink language and define properly the objective function subject to minimization within Matlab platform. The applied optimization procedure depends on the defined problem. Matlab is equipped with a rich variety of optimization procedures, able to solve either constrained or unconstrained minimization problems.

## 6. Conclusions

The paper discussed the application of Simulink in modeling the dynamical systems. It has been shown that Simulink enables to simulate in an easy way the dynamical behavior of the systems using either state space or transfer function description of the individual blocks forming the system under analysis. The efficiency of application of this approach has been illustrated on the example of the model of 3-phase induction machine and the linearized model of the power system directed to the stabilization of the operating frequency.

The other two examples have considered the non-typical application of Simulink at its cooperation with Matlab. The first application has been concerned with determination of the equilibrium points and their stability. In this approach the nonlinear system is modeled in Simulink and the other operations are done in Matlab. As an additional result of such application we may create also the phase portrait of the nonlinear system. The second example has presented the cooperation of Simulink with the optimization procedure of Matlab to solve the optimization problem aimed on minimizing the transient period in RLC electrical circuit.

Summarizing, we may conclude that Simulink is a very efficient and user friendly tool for modeling the dynamical systems. Thanks to very rich libraries and cooperation with other Matlab functions we can solve very complex problems concerned with the analysis and optimization of the linear and nonlinear dynamical systems.

## References

1. Banerjee S., Cahaterjee J. K. and Tripathi S. C.: Application of magnetic energy storage unit as load frequency stabilizer. IEEE Trans. Energy Conversion. Vol. 5, No 1, pp. 46-51, 1990.
2. Dabney J. and Harman T.: Mastering Simulink 2. Prentice Hall, Upper Saddle River, N.J., 1998.
3. Duffy D.: Mathematics for engineers with Matlab, CRC, Washington, 2003.
4. Eiali T. and Karim M., Continuous signals and systems with Matlab, CRC, Washington, 2001.
5. Elgerd O.: Electric energy systems theory, Intern. Student Edition, 1983 .
6. Karris S.: Signals and systems with Matlab applications, Wiley, N.Y., 2003.
7. Matlab – user manual, MathWorks, Natick, USA, 2010.
8. Matlab - Optimization toolbox, User manual, MathWorks, Natick, USA, 2010.
9. Nagase M., Ashino R. and Vaillancourt R.: Behind and beyond the Matlab ODE suite. report CRM-2651, 2000.
10. Osowski S.: Modeling and simulation of dynamical systems and processes (in Polish). Oficyna Wydawnicza PW, Warszawa, 2007.
11. . Osowski S., Cichocki A. and Siwek K.: Matlab in application to calculations of circuits and signal processing (in Polish). Oficyna Wydawnicza PW, Warszawa, 2006.
12. Simulink – user manual. MathWorks, Natick, USA, 2010

## SIMULINK ЯК ЕФЕКТИВНИЙ ІНСТРУМЕНТ АНАЛІЗУ ДИНАМІЧНИХ ЕЛЕКТРИЧНИХ СИСТЕМ

С. Осовські

Simulink є потужним набором інструментів Matlab-у, дуже корисним при аналізі динамічних систем. У статті показано застосування Simulink/Matlab до аналізу динамічних електричних систем. Показано, що це є ідеальним засобом для чисельного та якісного аналізу електричних та електронних мереж. Його ефективність показано на прикладі моделювання 3-фазної індукційної машини та енергетичної системи. Некласичні приклади використання Simulink-у включають аналіз нелінійної системи у просторі станів, визначення стійкості станів рівноваги, а також оптимізацію електричного кола. Слід зауважити, що сучасна версія Simulink-у дозволяє моделювати електричні мережі безпосередньо на підставі структури кола (спеціалізована бібліотека *SimElectronics*) без явного опису кола системою рівнянь.



**Stanislaw Osowski** was born in Poland in 1948. He received the M.Sc., Ph.D., and Dr. Sc. degrees from the Warsaw University of Technology, Warsaw, Poland, in 1972, 1975, and 1981, respectively, all in electrical engineering.

Currently he is a professor of electrical engineering at the Institute of

the Theory of Electrical Engineering, Measurement and Information Systems, Warsaw University of Technology and is also employed in Electronic Faculty of Military University of Technology, Warsaw, Poland. His research and teaching interest are in the areas of circuit theory, artificial neural networks, optimization techniques, and computer-aided system analysis and design. He is a Senior member of IEEE.