
Dynamic Simulation of Electrical Machines and Drive Systems Using MATLAB GUI

Viliam Fedák, Tibor Balogh and Pavel Záskalický

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/48519>

1. Introduction

Since the first appearance, the fields of electrical machine and drive systems have been continuously enriched by introduction of many important topics. Progress in power electronics, microcontrollers, new materials and advances in numerical modeling have led to development of new types of electrical machines and in field of electrical drives to realization of complex control algorithms. Their verification is usually done by simulation during system design, thus the effort is concentrated to development of simulation models.

MATLAB offers almost infinite possibilities for easy development of system models. MATLAB GUI (Graphic User Interface) in connection with Simulink and specialized toolboxes present a suitable and easy programmable tool for development of purpose-oriented virtual model of any dynamical system. Easy and comfortable change of parameters by control elements in MATLAB GUIDE (GUI Development Environment), such as push- and radio- buttons, text boxes, and easy visualization of results, enable to develop virtual models without deep knowledge of their substance nor without a tedious programming and debugging the models.

Well-elaborated models of electrical drives and machines available on-line were developed by (Riaz, n.d.). (Saadat, 2012) presented application of MATLAB GUI for electrical engineering subjects, available online and MATLAB GUI was utilized in (Petropol-Serb et al, 2007) for development of virtual model of induction machine.

Our contribution aims to present methodology and results in development of unified series of virtual models for electrical machines and drive systems using MATLAB GUI. The organization of the contribution is as follows: after brief description of tasks at virtual models design in GUI MATLAB (chapter 2) in the third chapter we describe development of few typical GUI oriented models of (more complex) electrical machines starting from a

simple outline of mathematic model, following by simulation model. Emphasis is put on development of the virtual model itself and description of its features. The fourth chapter deals with CAD of drive controllers using GUI MATLAB. Finally, in the fifth chapter we share some experiences from development of the GUIs and their utilization for training of students. In conclusion we also present ideas for our future work.

2. Design methodology for virtual models of electrical machines and drives

2.1. Tasks in design of the GUI screen

The GUI providing human-computer interaction presents one of the most important parts when working with the system model. User interacts with the computer easily, intuitively, without need for derivation, design, development, composition, and debugging the simulation model; without necessity to learn its operation, and finally, he gets required information in transparent, well-arranged form. In the fact, such GUI presents a functional virtual model, where the user sets system parameters, chooses mode of operation and required outputs to observe results. Design of GUI starts with careful planning of the following tasks:

1. Derivation of system mathematical model
2. Getting, debugging, and verification of simulation model
3. Programming GUI
4. Determination of input parameters changes (editing boxes, sliders)
5. Determination of outputs in graphical and text form
6. Design of the screen (or a set of interconnected screens)
7. Choice of calculation modes and algorithm of their control
8. Final refining and verification of functionality of the designed GUI screen

2.2. Principles of ergonomics of the GUI screen

When designing a functional GUI screen for the technical systems, designer must understand principles of good interface and screen design. Generally, the rules are described in (Galitz, 2007). We have adapted and extended them for design of virtual model – of a GUI MATLAB screen. The most important principles, when designing the placement of objects on the GUI screen, are:

- *Legibility* – saying that information should be distinguishable.
- *Facility* – how easy is the designed GUI screen intuitively usable.
- *Readability* – how information is identifiable and interpretable.
- *Attractivity* – to attract and call attention to different screen elements (placement of control elements and outputs, using colors, ...).
- *Guiding the eye* – by placement and grouping command objects by visual lines/boxes.

Further, designer should deal with user considerations, as follows:

- Visually pleasing (user friendly) *composition of the screen*.
- *Organizing screen elements* (balance, symmetry, alignment, proportion, grouping).
- Screen navigation and flow.
- Choice of implicitly *pre-setting system parameters* and their range (so that virtual models can be generally used in larger range of parameter changes).
- *Changing system parameters* by sliders or by numerical values in editing boxes.
- Finally, designer has to maintain *ergonomic of the screen* where the control elements and outputs should be organized in a legible way.

3. Virtual models for analysis of dynamical properties of electrical machines

In background of every GUI MATLAB there is working a simulation model of the system derived from its mathematical model. The same procedure is applied at development of GUI for electrical machines and drives. Let's show the GUI MATLAB development procedure on few electrical machines – the AC induction machine (asynchronous motor) and the brushless DC motor.

3.1. AC drive with 3-phase asynchronous motor

The AC drive consists of an AC machine supplied by a converter. The variables of AC machine (an asynchronous motor in our case) like electrical quantities (supply voltages and currents), magnetic variables (magnetic fluxes), and mechanical variables (motor torque and rotor angular speed) are usually to be investigated in:

- Various *reference frames* (rotating coordinate systems). In case of asynchronous motor two basic reference frames are considered:
 - $\{\alpha, \beta\}$ reference frame associated with stator, whose angular speed $\omega_k = 0$
 - $\{x, y\}$ reference frame rotating by synchronous angular speed $\omega_k = \omega_1$
- Various modes of supply:
 - harmonic (sinusoidal voltage)
 - non-harmonic (stepped voltage), PWM

3.2. Asynchronous motor model

For dynamic properties investigation of asynchronous motor (influence of non-harmonic supply to properties of the AC drive, etc.) a dynamical model of AC machine is used. The AC machine is described by set differential equations. For their derivation some generally accepted simplifications are used (not listed here) concerning physical properties, construction of the machine, electromagnetic circuit, and supply source.

In order to simplify mathematical model of the squirrel cage motor, the multiphase rotor is replaced by an equivalent three-phase one and its parameters are re-calculated to the stator. Equations describing behavior of the machine are transformed from three- to two-phase

system what yields to decreased number of differential equations. The quantities in equations are transformed into reference systems.

To derive dynamic model of asynchronous motor, the three-phase system is to be transformed into the two-phase one. In the fact, this transformation presents a replacement of the three-phase motor by equivalent two-phase one. The stator current space vector having real and imaginary components is defined by the equation:

$$\bar{\mathbf{i}} = \frac{2}{3}(i_a + \mathbf{a}i_b + \mathbf{a}^2i_c)$$

where

$$\mathbf{a} = e^{j120^\circ} = -\frac{1}{2} + j\frac{\sqrt{3}}{2}, \quad \mathbf{a}^2 = e^{j240^\circ} = -\frac{1}{2} - j\frac{\sqrt{3}}{2}$$

Basic equations of the AC machine with complex variables (denoted by a line over the symbol of the variable) in the reference frame rotating by general angular speed ω_k are:

$$\bar{u}_1 = R_1 \bar{i}_1 + \frac{d\bar{\Psi}_1}{dt} + j\omega_k \bar{\Psi}_1 \quad (1)$$

$$\bar{u}_2 = R_2 \bar{i}_2 + \frac{d\bar{\Psi}_2}{dt} + j(\omega_k - \omega)\bar{\Psi}_2 \quad (2)$$

$$\frac{J}{p} \frac{d\omega}{dt} = \frac{3p}{2} \text{Im}(\bar{\Psi}_{1k}^c \bar{i}_{1k}) - m_z \quad (3)$$

where the nomenclature is as follows:

- u_1, i_1, R_1, L_1 – voltage, current, resistance and inductance of stator phase winding
- u_2, i_2, R_2, L_2 – voltage, current and resistance and inductance of rotor phase winding (re-calculated to stator quantities)
- Ψ_1, Ψ_2 – total magnetic flux of the stator and rotor (re-calculated to stator side)
- ω – (rotor) mechanical angular speed
- m_z – load torque
- ω_k – angular speed of a general rotating reference frame $\omega_k = \omega_1$ or 0
- ω_1, ω – magnetic field angular speed, rotor angular speed, where $\omega = \omega_1 - \omega$
- σ – leakage factor $\sigma = (L_1 L_2 - L_h^2) / L_1^2$

For manipulation between various reference frames in the motor model the transformation formulas are used as listed in Tab. 1. All rotor parameters and variables are re-calculated to the stator side.

After inserting real and imaginary components into the complex of variables (e.g. for stator voltage $\bar{u}_1 = u_{1x} + j u_{1y}$ in synchronously rotating reference frame $\{x, y\}$), we get the AC motor

mathematical model whose equations are listed in Tab. 2 and a block diagram shown in Fig. 1 where $K_1 = 1/(\sigma L_1)$, $K_2 = 1/(\sigma L_2)$, $K = L_h / (\sigma L_1 L_2)$.

Transformation	Matrix notation	Block diagram
$\{a, b, c\} \rightarrow \{\alpha, \beta\}$ from 3-phase system $\{a, b, c\}$ to 2-phase reference frame $\{\alpha, \beta\}$ fixed with the stator (Clark transform)	$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$	
$\{\alpha, \beta\} \rightarrow \{a, b, c\}$ from 2-phase reference frame fixed with stator $\{\alpha, \beta\}$ into 3-phase system $\{a, b, c\}$ (inverse Clark transform)	$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix}$	
$\{x, y\} \rightarrow \{\alpha, \beta\}$ from synchronously rotating reference frame $\{x, y\}$ into the stationary frame $\{\alpha, \beta\}$ (Park transform)	$\begin{bmatrix} i_x \\ i_y \end{bmatrix} = \begin{bmatrix} \cos \rho & \sin \rho \\ -\sin \rho & \cos \rho \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix}$ $\rho = \omega_1 t$	
$\{\alpha, \beta\} \rightarrow \{x, y\}$ from stator reference frame $\{\alpha, \beta\}$ into the synchronously rotating frame $\{x, y\}$ (inverse Park transform)	$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} \cos \rho & -\sin \rho \\ \sin \rho & \cos \rho \end{bmatrix} \begin{bmatrix} i_x \\ i_y \end{bmatrix}$ $\rho = \omega_1 t$	

Table 1. Transformation relations between three-phase system and two-phase reference frame and between $\{x, y\}$ and $\{\alpha, \beta\}$ reference frames

	Magnetic fluxes	Relation between fluxes and currents
Stator	$\frac{d\psi_{1x}}{dt} = u_{1x} - R_1 i_{1x} + \omega_k \psi_{1y}$	$i_{1x} = K_1 \psi_{1x} - K \psi_{2x}$
	$\frac{d\psi_{1y}}{dt} = u_{1y} - R_1 i_{1y} - \omega_k \psi_{1x}$	$i_{1y} = K_1 \psi_{1y} - K \psi_{2y}$
Rotor	$\frac{d\psi_{2x}}{dt} = u_{2x} - R_2 i_{2x} - (\omega_k - \omega) \psi_{2y}$	$i_{2x} = K_2 \psi_{2x} - K \psi_{1x}$
	$\frac{d\psi_{2y}}{dt} = u_{2y} + R_2 i_{2y} - (\omega_k - \omega) \psi_{2x}$	$i_{2y} = K_2 \psi_{2y} - K \psi_{1y}$

Table 2. Equations of windings of asynchronous motor model in $\{x, y\}$ reference frame

The corresponding Simulink model is drawn in Fig. 1. The model of squirrel cage motor (rotor voltages = 0) contains 4 inputs and 10 outputs (Tab. 3).

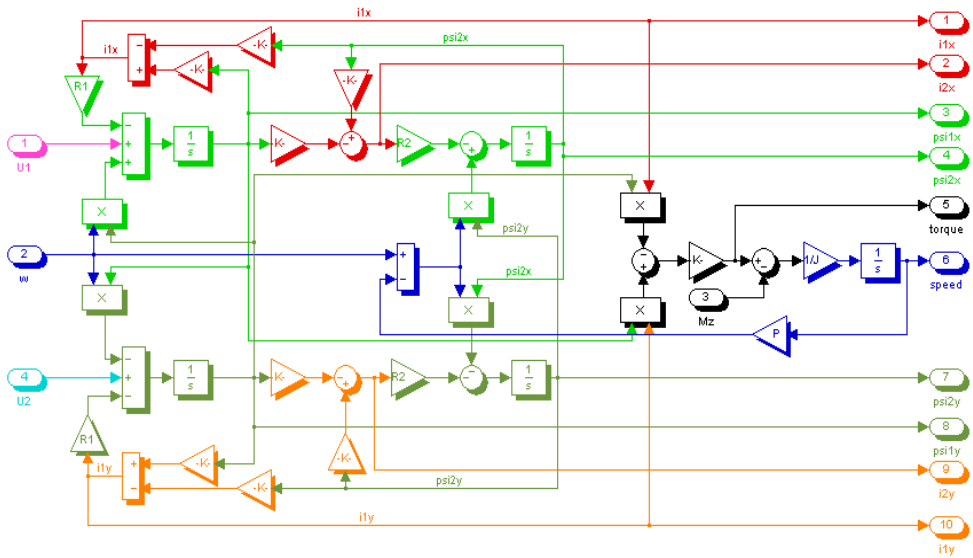


Figure 1. Simulink model of 3 phase squirrel cage asynchronous motor (the variables are denoted in the magnetic field reference frame $\{x, y\}$)

AM model inputs	AM model outputs
<ul style="list-style-type: none"> U_1 input voltage (axis x or α) U_2 input voltage (axis y or β) M_z load torque ω_k reference frame angular speed 	<ul style="list-style-type: none"> current \bar{i} (4 components) magnetic fluxes $\bar{\Psi}$ (4 components) motor torque M rotor angular speed

Table 3. Notation of inputs and outputs of the asynchronous motor model

3.2.1. Modeling of supply source

The asynchronous motor can be set into motion by various supply modes and control platforms:

- by *direct connection* to the supply network or to the frequency converter
- by *frequency starting* (with continuously increasing frequency of the supply voltage from the frequency converter)

Restrict our considerations to supply from indirect converter with the *Voltage-Source Inverter* (VSI). Based on the inverter control mode the output voltage can be:

- unmodulated (with 120 deg. switching in the power semiconductor devices)
- modulated by PWM

Developing inverter simulation schemes we have in mind two facts:

- the *constant stator flux* (i.e. fulfilling condition of constant ratio: $U_1/f_1 = \text{const.}$) should be preserved at all modes of motor control
- in range of very low frequency there should be kept an increased stator voltage (due to the voltage drop across the stator resistor) – so called V-curves (presenting a dependency of the supply voltage from the frequency). The V-curve can be modeled simply by a linear piecewise line.

The model of the motor supply source taking into consideration all described features is shown in Fig. 2 (signals denoted as SL and op are control signals from the GUI buttons). It has 4 inputs: supply frequency and voltage magnitude, ramp frequency and voltage (to simulate frequency starting). The switches “step/ramp” are controlled by pushbuttons from the GUI control panel.

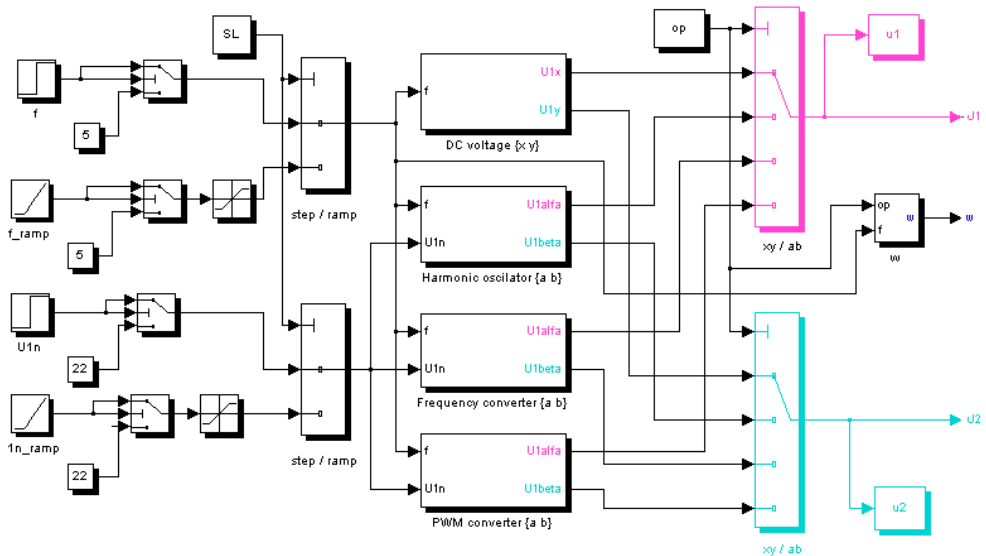


Figure 2. Simulink model of various modes of supply source (DC, harmonic, frequency converter and PWM)

Model of VSI converter (with constant output frequency)

We start to model the inverter output voltage based on a similarity of output converter voltage with the perpendicular harmonic voltages (Fig. 3a). The VSI voltage vector changes its position 6 times per period, after every 60° (Fig. 3b).

Proper switching instants are realized by comparators and switches (Fig. 4). Harmonic oscillator creates a core of the inverter model. Generation of six switching states during period of the output voltage is adjusted by comparing values of the \sin/\cos signals with preset values of $\sin 60 = \sqrt{3}/2$ for the voltage $u_{1\alpha}$ and value of $\cos 60 = 1/2$ for the voltage $u_{1\beta}$. The amplitudes of output voltage are adjusted by constants with values 1; 0,5 for $u_{1\alpha}$ and $0,866 = \sqrt{3}/2$ for $u_{1\beta}$.

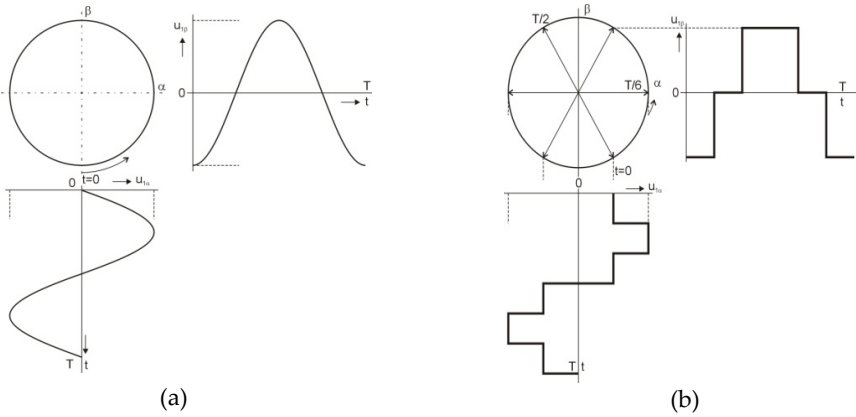


Figure 3. Simulink model of inverter

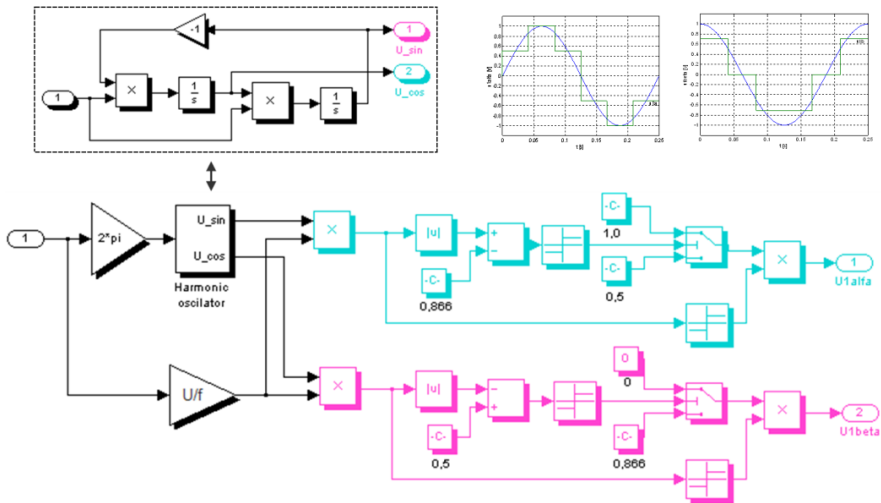


Figure 4. Simulation scheme realizing rectangle voltages $u_{1\alpha}$, $u_{1\beta}$ of the inverter

Model of PWM source

The simplest way to generate a PWM signal uses the interseptive method. The three-phase PWM voltage is generated directly in two axes $\{\alpha, \beta\}$ as shown in Fig. 5. The courses of the inverter PWM voltages $u_{1\alpha}$ and $u_{1\beta}$ are shown in Fig. 6. In frequency starting mode of the asynchronous motor, the frequency of supply voltage increases from zero to required final value. To get the stator flux constant, the voltage across the motor has to increase linearly with frequency ($U/f = \text{const.}$), except of very low frequency range (due to voltage drop across the stator resistor). For this purpose, the connection must be completed by a compensating circuit which increases the value of supply voltage keeping the ratio $U/f = \text{const}$ (Fig. 7). Up to the frequency of approx. 5 Hz the input voltage is kept constant on 10 % of its nominal value.

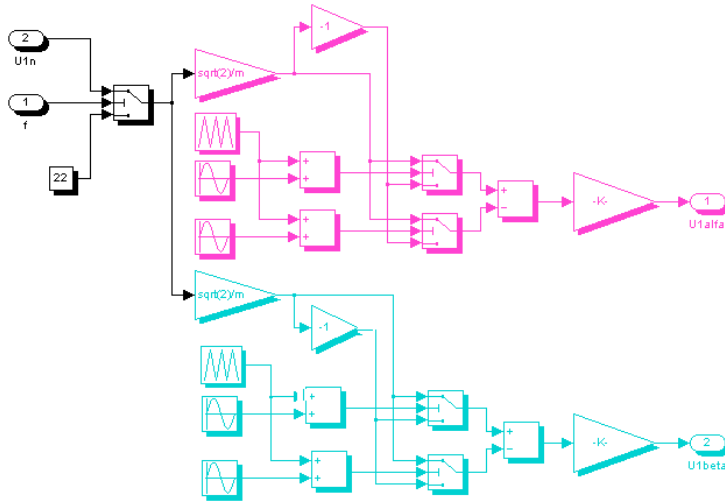


Figure 5. Model of voltages $u_{1\alpha}$ and $u_{1\beta}$ from the inverter with PWM

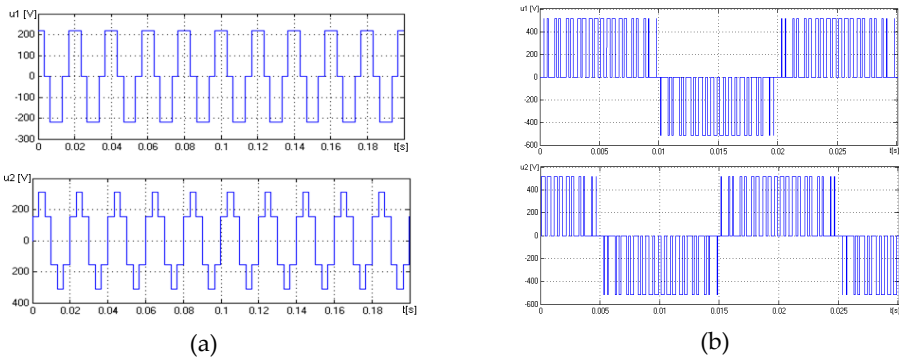


Figure 6. Output voltages and $u_{1\beta}$ and $u_{1\alpha}$ from the frequency converter: a) without and b) with PWM

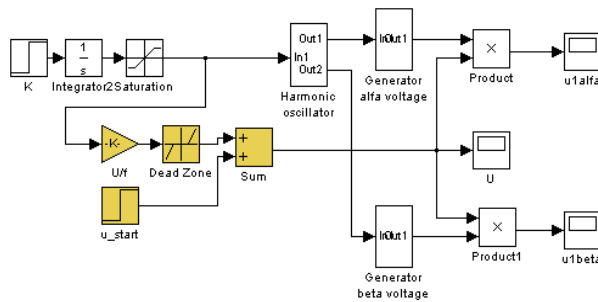


Figure 7. The model of converter realizing the frequency starting under consideration of the law of constant stator flux ($U/f = \text{const.}$)

The model supposes that amplitude of the DC link voltage is changed in the frequency converter. This solution is suitable for drives with low requirements to motor dynamics. The DC link contains a large capacitor what causes the DC link voltage cannot be changed step-by-step. The output inverter voltage can change faster if the PWM control is used. Output voltages of the inverter model with linear increasing frequency and voltage are shown in Fig. 8 (observe a non-zero amplitude of the voltage that at the starting what is consequence of described V-curve block).

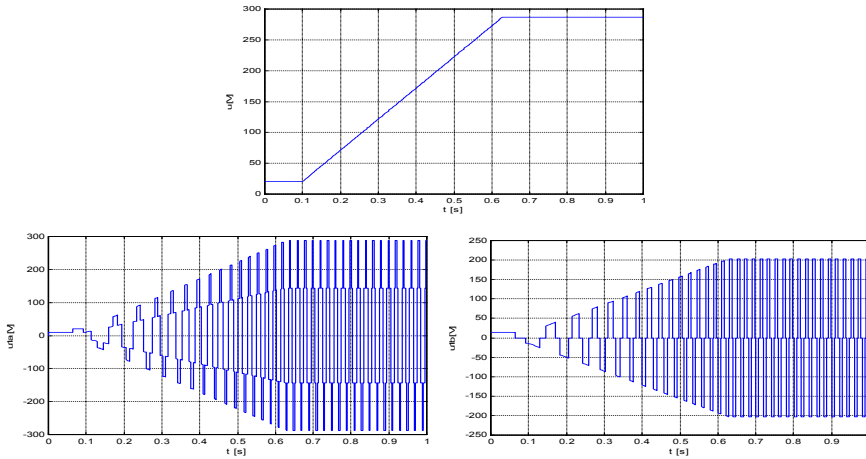


Figure 8. VSI output voltages $u_{1\alpha}$ and $u_{1\beta}$ at increased frequency (the frequency time course is on the top figure)

3.2.2. Model verification

The AC induction motor model was simulated using following motor parameters: $R_1=1,8 \Omega$; $R_2=1,85 \Omega$; $p=2$; $J=0,05 \text{ kgm}^2$; $K_1=59,35$; $K_2=59,35$; $K=56,93$.

Time courses of mechanical variables are shown in Fig. 10 (they are the same regardless the used reference frame). Motor dynamical characteristics $\omega = f(M)$ at various modes of supply are compared in Fig. 11.

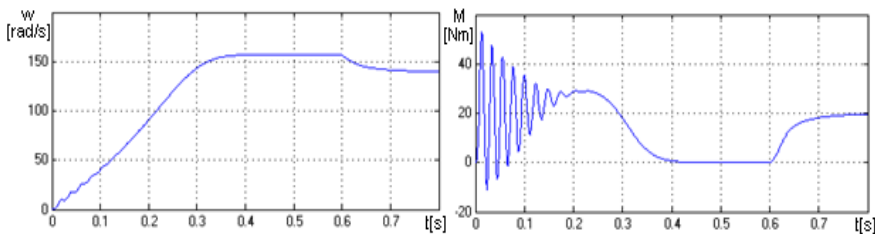


Figure 9. Time responses of asynchronous motor speed and torque at harmonic voltage supply at starting and loading the motor

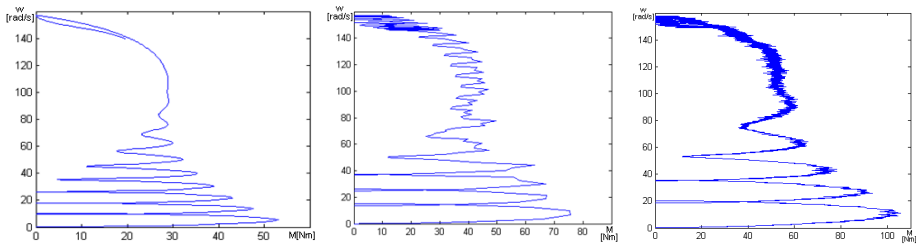


Figure 10. Dynamic characteristic of the asynchronous motor $\omega = f(M)$ supplied: a) by harmonic voltage, b) from frequency converter, c) from frequency converter with PWM

3.2.3. GUI design and realisation

After debugging the motor model (Fig. 11), development of GUI continues with careful design of the program flowchart and design of GUI screen.

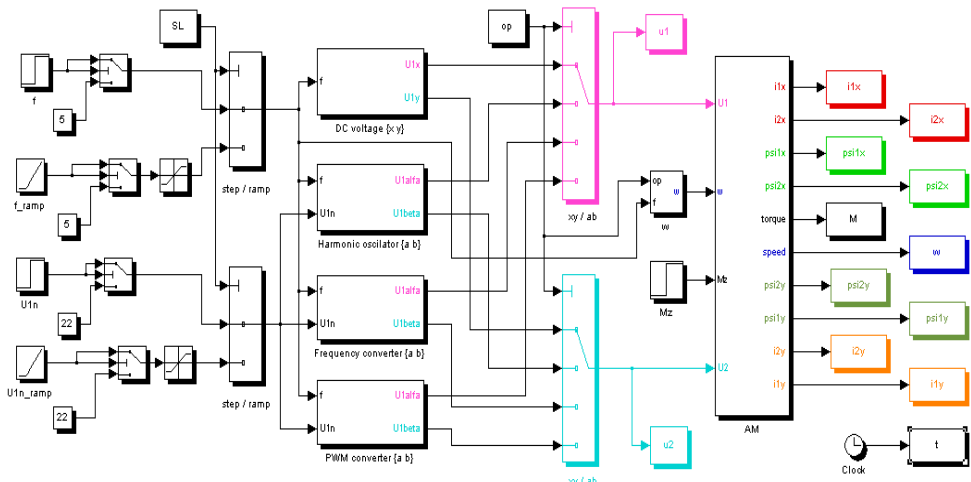


Figure 11. Arrangement of asynchronous motor subsystems in the Simulink GUI model

Description of the GUI functionality

The GUI screen (Fig. 13) consists of several panels. Their functionality is as follows:

- *Input data panel* (“Motor parameters”) in the bottom left part. The panel is used to system parameters entry. Their values can be changed by inserting numeric values into editing boxes. There is a possibility to return to original (default) parameters by pushing the button Default (in the pane *Mode*).
- Choice of *Coordinate reference frame system* (the panel on the right top part) enables to display motor output variables:
 - in the synchronously rotating reference frame
 - in the reference frame associated with the stator

- at harmonic supply
 - at nonharmonics supply from the VSI
 - at nonharmonics supply from the VSI with PWM
- *Output graphs.* Output variables are displayed in four graphs:
 - supply voltage time courses and in two coordinates
 - mechanical variables - motor torque and speed
 - stator currents or magnetic fluxes
 - rotor currents or magnetic fluxes
- The graph to be displayed can be chosen by pushing radio button in the menu *Graphs*. Time courses are chosen by the button *Time*; dependency of one variable on other is chosen by the button *Rectangular*.
- *Mode of starting* the motor can be selected in the panel *Motor supply*:
 - Direct connection to the supply – the button *Step*. The voltage U_1 (effective rms value) and frequency f_1 can be pre-set in the editing boxes.
 - Frequency starting – the button *Linear* enables to pre-set the frequency time rise starting from zero.
- Using the buttons in the panel *Mode* we start *Simulation*, at pressing *Default* (original) parameters are set, and the Simulink scheme is shown by pushing the button *Model*.

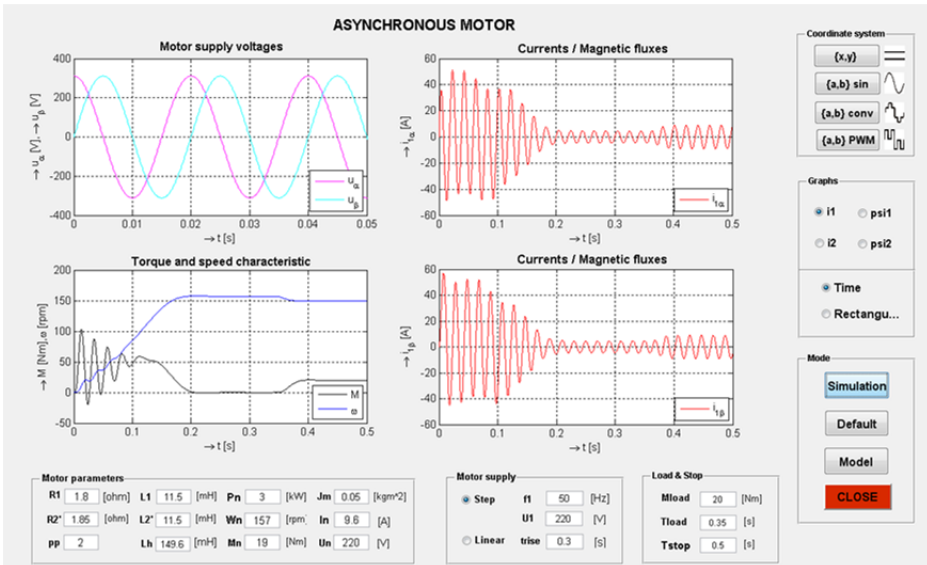


Figure 12. GUI screen of the AC drive with induction machine

Screen outputs

Samples of the screens displaying variables in the stator reference frame $\{\alpha, \beta\}$ are shown in Fig. 13:

- a. time courses at supplying motor by frequency converter – button *Time*)

- b. characteristics $M = f(\omega)$, $i_{1\alpha} = f(i_{1\beta})$, $\psi_{1\alpha} = f(\psi_{1\beta})$ - button Rectangular)
- c. time courses $i_{1\alpha} = f(t)$, $i_{2\alpha} = f(t)$ at supplying from the PWM frequency converter

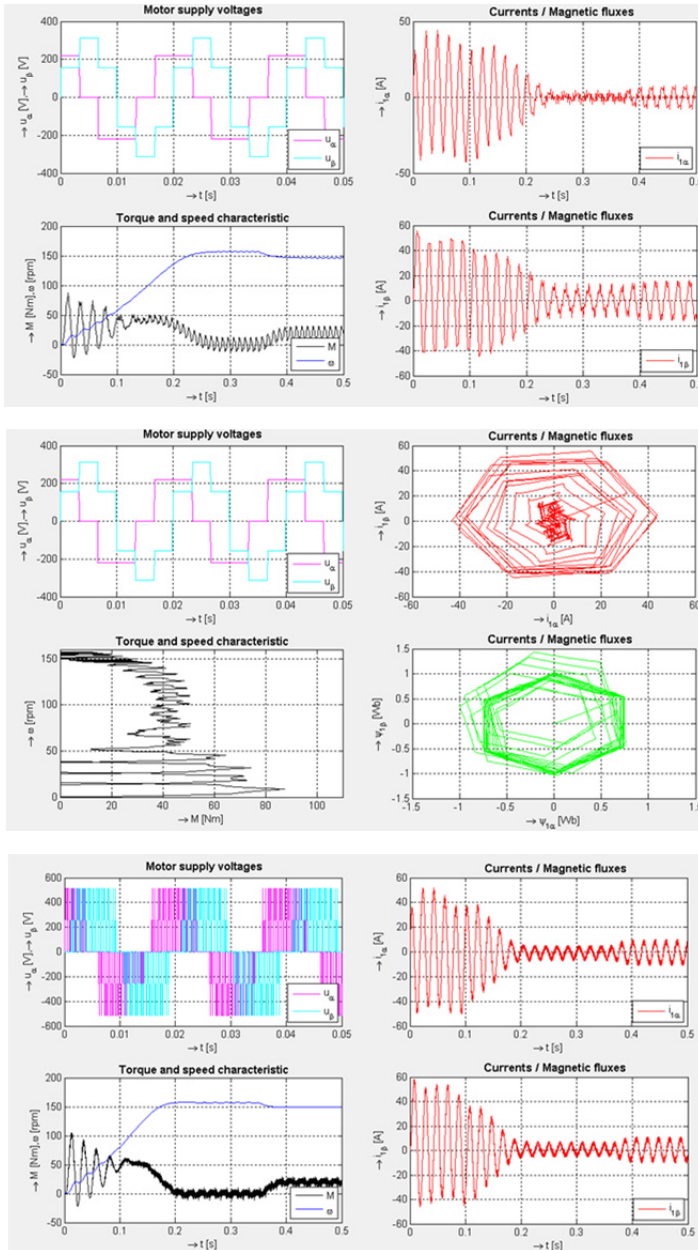


Figure 13. Examples of displaying various graphs in the GUI for asynchronous motor

3.3. BLDC motor

The Brushless Direct Current (BLDC) motor is rapidly gaining popularity by its utilization in various industries. As the name implies, the BLDC motor do not use brushes for commutation; instead of this they are commutated electronically.

The BLDC motors have many advantages over brushed DC motors and induction motors. A few of these are: (1) Better speed versus torque characteristics; (2) High dynamic response; (3) High efficiency; (4) Long operating life; (5) Noiseless operation; (6) Higher speed ranges. In addition, the ratio of torque delivered to the size of the motor is higher, making it useful in applications where space and weight are critical factors (Indu, 2008).

The torque of the BLDC motor is mainly influenced by the waveform of back-EMF (the voltage induced into the stator winding due to rotor movement). Ideally, the BLDC motors have trapezoidal back-EMF waveforms and are fed with rectangular stator currents, which give theoretically constant torque. However, in practice, a torque ripple exists, mainly due to EMF waveform imperfections, current ripple, and phase current commutation. The current ripple follows up from PWM or hysteresis control. The EMF waveform imperfections result from variations in the shapes of slot, skew and magnet of BLDC motor, and are subject to design purposes. Hence, an error can occur between actual value and the simulation results. Several simulation models have been proposed for analysis of BLDC motor (Jeon, 2000).

3.3.1. Construction and operating principle

The BLDC motor is also referred to as an electronically commuted motor. There are no brushes on the rotor and the commutation is performed electronically at certain rotor positions. In the DC commutator motor, the current polarity is reversed by the commutator and the brushes, but in the brushless DC motor, the polarity reversal is performed by semiconductor switches which are to be switched in synchronization with the rotor position. Besides of the higher reliability, the missing commutator brings another advantage. For the DC brushed motor the commutator presents also a limiting factor in the maximal speed. Therefore, the BLDC motor can be employed in applications requiring high speed (Jeon, 2000).

The BLDC motor is usually considered as a three-phase system and thus it has to be powered by a three-phase power supply. The rotor position must be known at certain angles, in order to align the applied voltage with the back-EMF. The alignment between the back-EMF and commutation events is very important.

A simple motor model of BLDC motor consisting of a three-phase power converter and a brushless DC motor is shown in Fig. 14.

3.3.2. Mathematical model of the BLDC motor

Modeling of a BLDC motor can be developed in the similar manner as a three-phase synchronous machine. Since there is a permanent magnet mounted on the rotor, some

dynamic characteristics are different. Similarly, the model of the armature winding for the BLDC motor is expressed as follows:

$$u_a = Ri_a + L \frac{di_a}{dt} + e_a \quad (4)$$

$$u_b = Ri_b + L \frac{di_b}{dt} + e_b \quad (5)$$

$$u_c = Ri_c + L \frac{di_c}{dt} + e_c \quad (6)$$

where L is armature self-inductance, R - armature resistance, u_a, u_b, u_c - terminal phase voltages, i_a, i_b, i_c - motor input currents, and e_a, e_b, e_c - motor back-EMF.

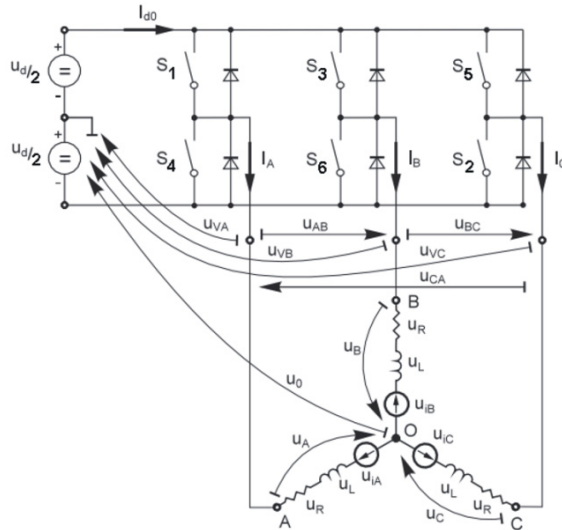


Figure 14. BLDC motor model

In the three-phase BLDC motor, the back-EMF is related to a function of rotor position and the back-EMF of each phase has 120° phase angle difference so the equation for each motor phase is as follows:

$$e_a = K_w f(\theta_e) \omega \quad (7)$$

$$e_b = K_w f(\theta_e - 2\pi/3) \omega \quad (8)$$

$$e_c = K_w f(\theta_e + 2\pi/3) \omega \quad (9)$$

where K_w is back EMF constant of one phase, θ_e - electrical rotor angle, ω - rotor speed. The electrical rotor angle θ_e is equal to the mechanical rotor angle θ_m multiplied by the number of poles p :

$$\theta_e = \frac{p}{2}\theta_m \tag{10}$$

Total torque output T_e can be represented as summation of that of each phase:

$$T_e = \frac{eA^iA+eB^iB+eC^iC}{\omega} \tag{11}$$

The equation of mechanical part is represents as follows:

$$T_e - T_l = J \frac{d\omega}{dt} + b\omega \tag{12}$$

where T_l is load torque, J - rotor inertia, b - friction constant.

3.3.3. Simulink model of the BLDC motor

Fig. 16 shows the block diagram of the BLDC motor SIMULINK model in the rotor reference frame.

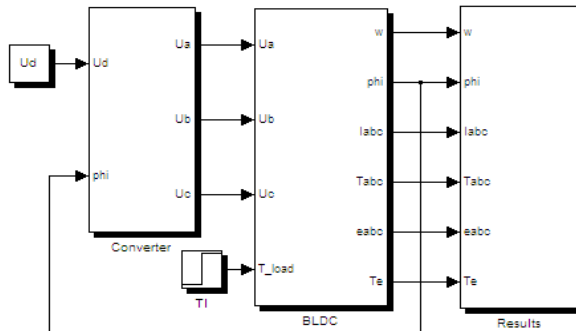


Figure 15. Simulink model of the BLDC motor

Fig. 16 shows detail of the BLDC motor block. Fig. 17a shows Simulink diagram of trapezoidal back-EMF and in Fig. 17b there is Simulink model of sinusoidal back-EMF. The trapezoidal functions and the position signals are stored in lookup tables that change their output according to the value of the electrical angle (Indu, 2008).

Unlike a brushed DC motor, the commutation of a BLDC motor is controlled electronically. To rotate the BLDC motor, the stator windings should be energized in sequences. In order to understand which winding will be energized following the energizing sequence, it is important to know the rotor position. It is sensed using Hall Effect sensors embedded into the stator. Most of the BLDC motors contain three Hall sensors embedded into the stator on the non-driving end of the motor. The number of electrical cycles to be repeated to complete a mechanical rotation is determined by rotor pole pairs. Number of electrical cycles/rotations equals to the rotor pole pairs. The commutation sequences are shown in Tab. 4.

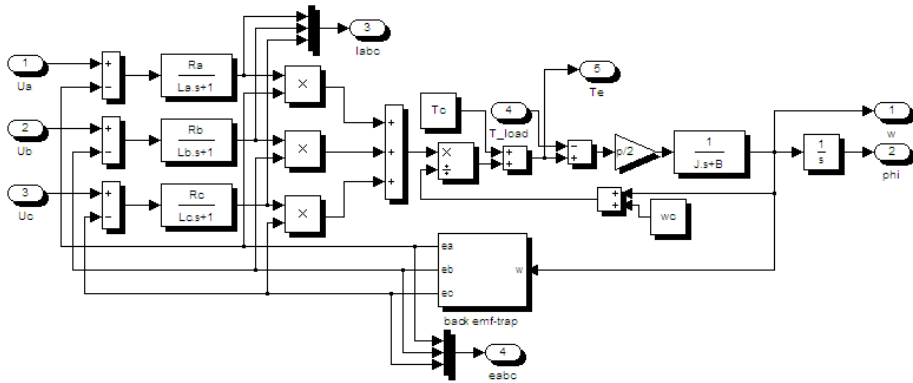


Figure 16. Detailed overview of the BLDC motor block

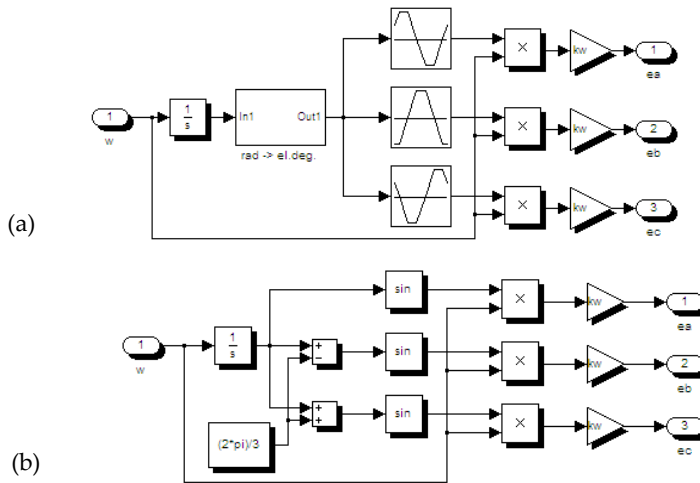


Figure 17. Trapezoidal (a) and sinusoidal (b) model of the back-EMF

Electrical degree	Hall sensor value (ABC)	Phase	Switches
0° - 60°	101	A-C	S1-S2
60° - 120°	001	B-C	S2-S3
120° - 180°	011	B-A	S3-S4
180° - 240°	010	C-A	S4-S5
240° - 300°	110	C-B	S5-S6
300° - 360°	100	A-B	S6-S1

Table 4. Electrical degree, Hall sensor value and corresponding commuted phase in clockwise rotation of the rotor

3.3.4. Mathematical and simulink model of the three-phase converter

The converter supplies the input voltage for three phases of the BLDC motor. Each phase leg comprises two power semiconductor devices. Fig. 18 shows the scheme of the considered three-phase converter.

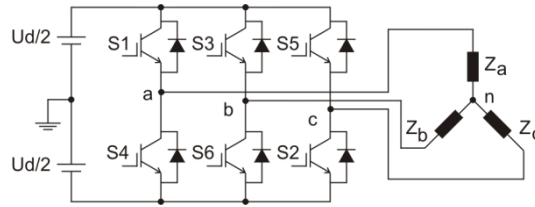


Figure 18. Modelled three-phase converter

Appropriate pairs of the switches (S1 to S6) are driven based on the Hall sensors input. Three phases are commutated in every 60° (el. degrees). The model of the converter is implemented using the equations:

$$U_{an} = S_1 \frac{U_d}{2} - S_4 \frac{U_d}{2} - U_f \tag{13}$$

$$U_{bn} = S_3 \frac{U_d}{2} - S_6 \frac{U_d}{2} - U_f \tag{14}$$

$$U_{cn} = S_5 \frac{U_d}{2} - S_2 \frac{U_d}{2} - U_f \tag{15}$$

where U_{an} , U_{bn} , U_{cn} are line-neutral voltages, U_d – the DC link voltage, U_f – the forward diode voltage drop.

Fig. 19a shows the Simulink model of the three-phase converter block. In the simulation we assumed an ideal diode with neglected voltage drop U_f . The *Commutation sequences* block was developed based on the commutation sequence shown in Tab. 4. Converter voltage waveforms that are switched according to the commutation sequences in Tab. 4 are shown in Fig. 19b.

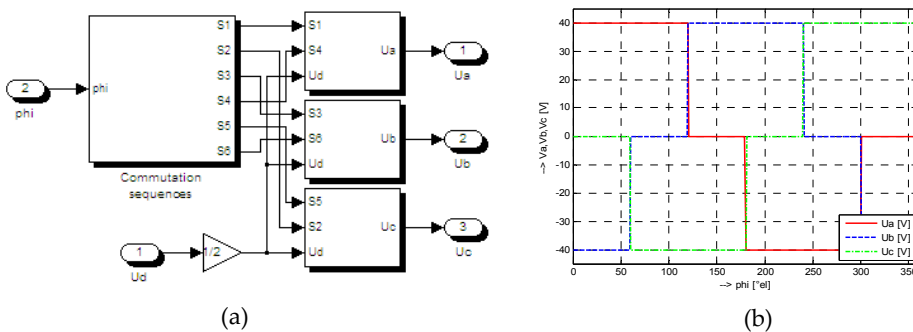


Figure 19. Detailed overview of the three-phase converter (a) and voltage source waveforms (b)

3.3.5. GUI of the BLDC motor

The simulated BLDC motor is presented in a graphical user interface GUI (Fig. 20).

By the buttons in the panel *Mode* we start the *Simulation*, put *Default* (original) values and show the Simulink *Model*.

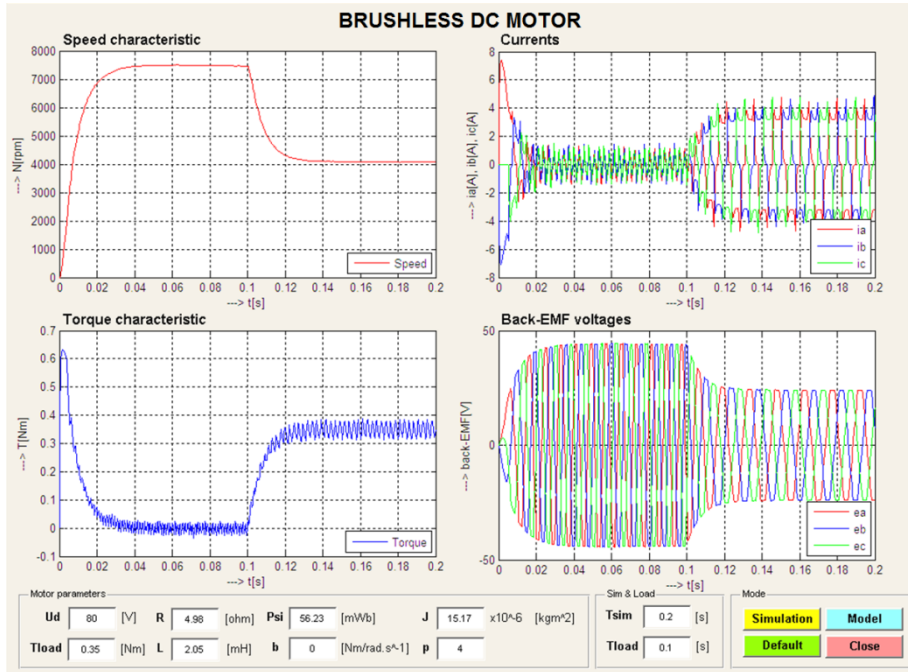


Figure 20. GUI for the BLDC motor

The default parameters of the BLDC motor for simulation are: $U_d=80$ V, $T_l=0,35$ Nm, $R=4,98$ Ω , $L=2,05$ mH, $\psi=56,23 \cdot 10^{-3}$ Wb, $b=0$ Nm/rad.s⁻¹, $J=15,17 \cdot 10^{-6}$ kgm², $p=4$.

4. Virtual models applied for synthesis of drive systems

MATLAB GUI presents an extremely suitable tool for development of models to support CAD design of drive controllers, whose algorithms are known. Two simple cases are presented below: design of controllers in the frequency and time domains, other cases are mentioned in the subchapter 5.1.

4.1. CAD design of controller parameters for DC motor drive in frequency domain

The DC drive controllers in the *frequency domain* are calculated mostly by using the following criteria:

1. The current controller of the PI type is calculated on basis of the *Optimum Modulus Criterion* (OMC) from the drive system parameters:

$$F_{Ri} = K_{Ri} + \frac{1}{sT_{ii}} \tag{16}$$

2. After calculation of the current controller parameters and current control loop simplification, the speed controller is calculated based on the *Symmetrical Optimum Criterion* (SOC). It is again of PI type having the transfer function:

$$F_{R\omega} = K_{R\omega} + \frac{1}{sT_{i\omega}} \tag{17}$$

Fig. 22 shows the principal block diagram of the system and in Fig. 23 there is view on the virtual GUI model of speed controlled DC drive.

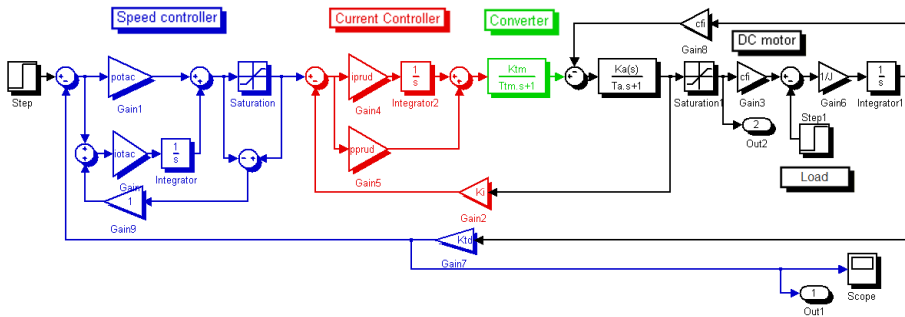


Figure 21. Control circuit of DC motor drive with current and speed controllers

Virtual model features

The user has a possibility to tune controller parameters in each design step according to displayed time response. The GUI screen consists of several panels:

- *Time response* - the graph with time courses of the motor current and speed. Immediately after change of any system parameter (motor -, drive -, or controller parameters) by a slider or inserting a numeric value into editable box the simulation starts and new time responses are drawn (like in a real drive).
- *Block diagram* - displays the block diagram of the system
- *System parameters* are changed by sliders or inserting values into the boxes.

Before starting the model, implicit parameters are set up, but they can be changed later. After pushing the button *Computed value* the parameters of controllers are calculated from the actual values of parameters. Simultaneously a small window appears there with a question whether the calculated values of controller parameters are acceptable or not (if not, user can set up own parameters and can to tune them according to the time responses of the drive). To return to starting values, the user pushes the button *Default* (similar to the system restart).

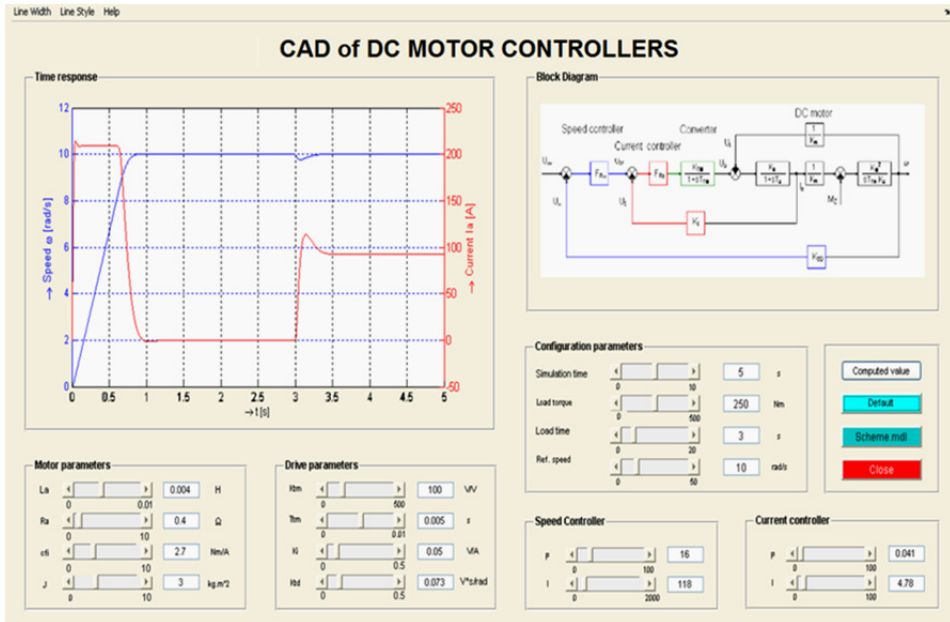


Figure 22. GUI screen for designing DC motor drive controllers in the frequency domain

4.2. CAD design of controller parameters for DC motor drive in time domain

The computing algorithm is different from calculation of the controllers in the frequency domain and the task belongs to more complex one. The computation starts from the state-space model of the DC motor having two inputs in one output, in the form of state equations:

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} + \mathbf{b} \cdot u + \mathbf{e} \cdot z = \begin{bmatrix} 0 & \frac{K_m}{T_m \cdot K_a} \\ -\frac{K_a}{K_m \cdot T_a} & -\frac{1}{T_a} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{K_T \cdot K_a}{T_a} \end{bmatrix} u + \begin{bmatrix} -\frac{K_m^2}{T_m \cdot K_a} \\ 0 \end{bmatrix} M_z \quad (18)$$

$$y = \mathbf{c}^T \mathbf{x} = [1 \ 0] \mathbf{x} \quad (19)$$

where \mathbf{A} is system matrix, \mathbf{x} – state vector, \mathbf{b} – input vector, \mathbf{c}^T – output (row) vector \mathbf{e} –disturbance vector, u – input variable, y – output variable.

The final control structure with the feedback through the state controller vector \mathbf{r}^T is clear from the Simulink model Fig. 23. The integrator at the input serves to reject constant or slowly changing disturbances what is a common case.

The state control structure parameters: K_1 , r_1 and r_2 are designed by known *pole placement method* where for a prescribed position of poles the required polynomial is compared with

the system polynomial and missing parameters of the controller are calculated from a set of linear algebraic equations.

The control structure in Simulink to simulate the system is shown in (Fig. 23).

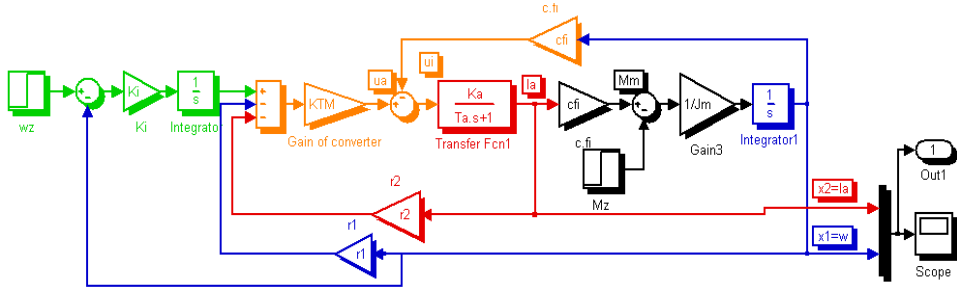


Figure 23. Simulink model of the state-space control of DC drive

Fig. 24 shows the GUI screen of the virtual model that enables to calculate state-space controller parameters and visualize time responses of the current and speed. It is a more complex GUI involving synthesis of the state-space controllers and giving the possibility to tune theoretically calculated parameters.

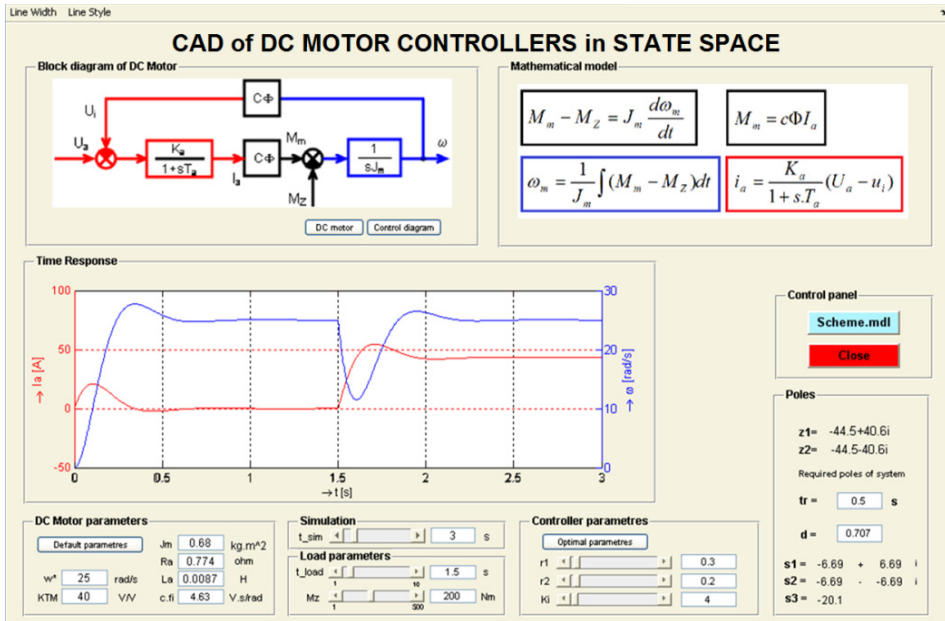


Figure 24. GUI screen for designing DC motor drive controllers in the state space domain

The panel *Controller parameters* serves to setting parameters of the state controller – by tuning or selecting the button *Optimal parameters* to calculate poles position placement.

Here:

- r_1 – feedback from state variable x_1 (motor speed),
- r_2 – feedback from state variable x_2 (motor current),
- K_i – gain of the integrator (to reject steady-state disturbances).

The state controller parameters are calculated automatically on basis of required values of control time and damping (panel *Poles*, the item *Required poles of the system*). In the upper part of the panel the real positions of poles are shown.

5. Experiences with utilization of virtual models

5.1. Utilization of virtual dynamical models

Except of the presented GUI screens of virtual models a series of tens other models from fields of electrical engineering and mechatronic systems was developed to suit institutional needs. They cover topics from Electrical Machines, Power Electronics, Electrical Actuators and Drives, Servodrives, Mechatronic Systems, Control Theory, and others. We have also developed some more complex GUIs, reported e.g. by (Ismeal & Fedák, 2012), calculating artificial intelligence algorithms - to design PID controllers using fuzzy logic and genetic algorithms with various objective functions to evaluate the best PID controller.

The developed GUIs serve as virtual models to clarify phenomena and enhance features of the systems during lectures, and to prepare students for laboratory work. In order students to get more skills and practical experiences prior entering lab their work consists of two phases:

Design and simulation – for a given system motor or drive a student has:

- to derive mathematical model,
- to compose the block diagram,
- to design control law and controllers (in case of drives),
- to verify system behavior by simulation.

Verification and analysis – a student has:

- to verify the design using a virtual reference model,
- to perform system analysis (“to play himself” with the virtual model) in order to investigate system behavior at various values of system parameters and in various working points (small experiments round working point).

5.2. Application of MATLAB compiler in virtual model development

A disadvantage of using GUI MATLAB consists in the fact, that the program can run only on a computer having installed the MATLAB program (and appropriate toolboxes containing instructions that are used in the main GUI program). This disadvantage can be suppressed by development of an executable (.exe) file from the original program. In this

case the developer must install the Compiler Runtime program. The MATLAB Compiler creates a standalone executable file from the MATLAB code, which can then run in a runtime engine called *Matlab Compiler Runtime* (MCR). Once compiled, the standalone application, along with MCR, can be shared with other users for free. The only problem is that the original GUI MATLAB program cannot contain any Simulink model (mdl file). As the GUI MATLAB systems usually contain Simulink models, they have to be replaced by MATLAB programs solving mathematical model by various algorithms.

6. Conclusion

The chapter describes principles and methodology of virtual models development in GUI MATLAB for few chosen electrical machines and controlled drives. The models perform analyses of real machines and drives in various working points and they enable easily to show system performance in various working points and to analyze influence of variable system parameters, modes of supply, and control parameters to system behavior. Presented virtual models have been of various complexity – the simpler ones enable virtual analysis of electrical machines and more complex virtual models also deal with algorithms for synthesis of drive controllers.

Strong advantage of developed virtual models consists in the fact user does not need to know the complexity of dynamical system whose simulation scheme is working in the background. He changes only system parameters, selects input signals (shape and amplitude of reference values, forcing and load signals), select mode of calculations and outputs (graphs displaying). The parameters of virtual models can be changed by a slider or by editing numerical values in editable boxes.

Based on the procedure a whole series of virtual models designed in GUI MATLAB has been developed at the authors' institution in recent years which are partly accessible through the website of the Virtual Laboratory for Control of Mechatronic Systems (KEM TU Kosice, 2010). The virtual dynamical models contribute to e-learning support at teaching and they also serve for preparation of students for laboratory experimentation. Their utilization makes more attractive lectures and considerably enhances explanation of systems properties. By simulation model students generally easier understand physical processes and they are better prepared to the laboratory work.

Shortcoming of developed models consists in the fact they run on computers having installed the MATLAB program. To overcome this, recently our work was concentrated to applications of the MATLAB Compiler to develop executable files. It should be noted that this application enables to run MATLAB operation without simulation (i.e. without a Simulink scheme), without 3D virtual reality views and without animation. The advantage on one side makes development of virtual models more difficult using more complex mathematical subroutines. Also some known problems with GUI MATLAB should be noted - cross platform appearance may not be the same and during the GUI development, often must be used tricks and somehow unfriendly techniques.

Author details

Viliam Fedák and Pavel Záskalický
Technical University of Košice, Slovakia

Tibor Balogh
Magneti Marelli, Electronic Systems Division, Industrial Park Kechnec, Slovakia

Acknowledgement

The financial support of the Slovak Research and Development Agency under the contract No. APVV-0138-10 is acknowledged. The work was also supported by Slovak Cultural and Educational Agency of the Ministry of Education of Slovak Republic under the contract KEGA 042TUKE-4/2012 "Teaching Innovation in Control of Mechatronic Systems".

7. References

- Galitz, W. O. (2007). *The Essential Guide to User Interface Design. An Introduction to GUI Design Principles and Techniques*, Wiley Publishing, Inc., ISBN 978-0-470-05342-3, Indianapolis, Indiana
- Hill, S. (2011). How to Make an Executable From MATLAB Code, Date of access: March 31, 2011, <http://www.ehow.com/how_8686360_make-executable-matlab-code.html>
- Indu, B., Ashly, R. & Tom, M. (2008). Dynamic Simulation of Brushless DC Drive Considering Phase Commutation and Backemf Waveform for Electromechanical Actuator. *IEEE TENCON*, Hyderabad 2008. ISBN: 978-1-4244-2408-5
- Ismeal G.A. & Fedák V. (2012). Overview of Control Algorithms for DC Motor Drive as a Basis for Development of GUI in MATLAB. *XXVI. microCAD International Scientific Conference*, Miskolc, 2012, 9 p., ISBN 978-963-661-773-8
- Jeon, Y.S., Mok, H.S., Choe, G.H., Kim, D.K. & Ryu, J.S. (2000). A New Simulation Model of BLDC Motor with Real Back EMF waveforms. *IEEE CNF on Computers in Power Electronics*, COMPEL 2000, pp. 217 – 220, July 2000
- KEM TU Kosice (2010). Virtual Laboratory of Mechatronic Systems Control. , In: Date of access: March 31, 2012, Available from <<http://andromeda.fei.tuke.sk/>> (in Slovak)
- MathWorks (n.d.). Deploytool - R2012a Documentation MATLAB Compiler
<http://www.mathworks.com/help/toolbox/compiler/deploytool.html>
- Parspour, N. & Hanitsch, R. (1994). Fuzzy Controlled Brushless DC Motor For Medical Applications. *Industrial Electronic, Control and Instrumentation IECON, IEEE*, Bologna, 1994, ISBN: 0-7803-1328-3
- Petropol-Serb, G.D.; Petropol-Serb, I.; Campeanu, A. & Petrisor, A. (2007). Using GUI of Matlab to create a virtual laboratory to study an induction machine. *EUROCON*, 2007. *The International Conference on Computer as a Tool*, ISBN 978-1-4244-0813-9, Warsaw, September 9-12, 2007
- Riaz, M. (n.d.): Simulation of Electric Machine and Drive Systems Using MATLAB and Simulink. University of Minnesota. www.ece.umn.edu/users/riaz/

Saadat, H. (2012). MATLAB Graphical User Interface for EE Students. Date of access: March 31, 2012, Available from <<http://people.msoe.edu/~saadat/matlabgui.htm>