Software and hardware solution for digital signal processing algorithms testing

M. V. Talanov¹*, and V. M. Talanov¹
¹Ogarev Mordovia State University, Saransk, Russia

Abstract. The article describes the microprocessor system for various digital signal processing algorithms testing. The development of electric drive control systems is carried out with the usage of modeling systems such as, MATLAB/Simulink. Modern digital control systems are based on specialized digital signal microcontrollers. The present market offers evaluation boards, for example STM32F4DISCOVERY, which enables to connect a microcontroller to a personal computer. It makes possible to use the microcontroller as a part of the mathematical model of the control system. However, the designing of the control system simulation model and the program for the microprocessor is carried out in different programming environments. Thus, the software and hardware solution for testing programs for the microprocessor, which is a part of the control system, is relevant. This article deals with the designing of the modeling method in which the prototype program for the microprocessor is debugged as a part of the electric drive control system simulation model.

1 Introduction

A microcontrollers flash memory, which is designed to store programs, has a limited number of overwriting cycles. For example, 10,000 cycles for the STM32 series of microcontrollers. Thus, it is advisable firstly to design a microcontroller program prototype on a personal computer and then to prepare the final program for the microcontroller on the basis of the debugged prototype in a specialized development environment [1], [2]. The paper proposes a hardware and software solution based on evaluation board STM32F4DISCOVERY with microcontroller STM32F407 that enables: to design and test the operation of microprocessor program prototypes; to integrate a microprocessor in the electric drive control system simulation model; to perform simulation of the electric drive control system; to process the control system simulation results. It is considered in the paper that a rotor speed estimation of an induction motor by the extended Kalman filter (EKF) tests the proposed solution.

2 The software and hardware solution structure

A microcontroller program prototype is a software implementation of the main data processing algorithm without additional initialization of the microcontroller periphery. MEX (Matlab Executable) files enable to call C subroutines from the MATLAB programs. Thus, it is possible to test source code fragments designed for the microcontroller. Once all the fragments are combined together with the initialization of the microcontroller auxiliary periphery into a single program, it can be loaded into the microcontroller memory and the developed solution enables to embed the microcontroller in the electric drive control system model in Simulink. The STM32F4DISCOVERY evaluation board enables to connect an auxiliary equipment to test different aspects of the electric drive control system mathematical model.

The functional diagram of the proposed solution is shown in figure 1.

Fig. 1. Functional diagram of the proposed solution.

The developed C# form application enables to configure the operating mode of the control system model. The form is used to set the connection parameters to the microcontroller, if it is included in the mathematical model. When you select a MEX function with the required data processing program, the parameters of the selected algorithm are specified in the form. The results of data processing are recorded in the
database during the modelling process. The graphical form uses the .NET component compiled in MATLAB to plot the data written to the database during the simulation.

The flow diagram of the suggested solution is shown in figure 2.

A Microsoft .NET Framework is a software platform which enables to combine a code written in different programming languages into a single software package. In this case, the programs for the microcontroller are made in C, and the graphical shell are made in C#. The C# language was created for programming in the .NET environment and supports all its capabilities. The C# language is an object-oriented programming language that enables to create web applications, graphic forms, to work with various database management systems. It is possible to use .NET Assemblies in a code in MATLAB. In the proposed solution .NET Assembly includes a graphical form that designs objects to configure the solution and controls the operation of the Simulink model. It is possible to design function pointers in C language. It is possible to use this feature by passing some object method call to the MEX function. Next, we dereference the pointer to the passed function in the C program, and thus can control the objects designed in the C# program. The proposed approach will combine the capabilities of C, C#, MATLAB languages and expand the modelling capabilities through the use of various methods of object-oriented programming. Figure 3 shows the programming technologies used in the proposed software and hardware solution.

3 Test example

To test the proposed solution let’s consider the mathematical model of the extended Kalman filter [3], [4], [5], which is used in the control system to determine the rotor speed estimation of an induction motor [6], [7].

4 The EKF implementation

Structure of the EKF is based on the mathematical model of the induction machine expressed in the stationary reference frame [8], [9].

Fig. 2. Flow diagram of the suggested solution.
The system state vector is $x = [i_{s\alpha}, i_{s\beta}, \psi_{r\alpha}, \psi_{r\beta}, \omega_r]^T$; the starting value of the state vector is $x_0$; the estimation of the state vector is $\hat{x}$; the input system is $u = [u_{s\alpha}, u_{s\beta}]$; the zero-mean white Gaussian noises with covariance $Q$ and $R$ are $v$ and $w$; $A_d$ – a discrete system matrix; $B_d$ – a discrete input matrix; $C_d$ – a discrete output matrix. Matrices $A_d$, $B_d$, and $C_d$ are:

$$A_d = \begin{bmatrix}
1 - \frac{T}{K_s} & 0 & \frac{L_{m\alpha}}{K_s} & 0 & \frac{T}{K_s} \\
0 & 1 - \frac{T}{K_s} & -\frac{L_{m\alpha}}{K_s} & 0 & -\frac{T}{K_s} \\
0 & 0 & 1 - \frac{T}{K_s} & 0 & -\frac{T}{K_s} \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}$$

$$B_d = \begin{bmatrix}
\frac{T}{L_s'} & 0 & 0 & 0 & 0 \\
0 & \frac{T}{L_s'} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$C_d = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

and $L_s' = \sigma L_s$ – a transient stator inductance, where $\sigma = 1 - L^2_m / (L_s L_r)$ – a leakage coefficient, $L_s$ – a stator inductance; $L_m$ – a magnetizing inductance; $L_r = L_{ip} + L_m$ – a rotor inductance, where $L_{ip}$ – a rotor leakage inductance; $T_r = L_r / R_r$ – a rotor time constant; $\omega_r = \omega_m p$ – an electrical angular rotor speed, where $\omega_m$ – a mechanical angular rotor speed, $p$ – a number of pole-pairs; $K_s = L_s' / (R_s + R_r (L_m / L_r)^2)$ – a parameter, where $R_s$ – a stator resistance, $R_r$ – a rotor resistance; $i_{s\alpha}$ and $i_{s\beta}$ – stator currents in the stationary reference frame, $u_{s\alpha}$ and $u_{s\beta}$ – a linear stator voltages in the stationary reference frame, $\psi_{r\alpha}$ and $\psi_{r\beta}$ – rotor flux-linkages in the stationary reference frame.

The EKF algorithm contains two stages: extrapolation – equations (1)-(2), and correction – equations (3)-(5):

$$\hat{x}(t_{k+1}) = F(\hat{x}_k)P(t_k)F^T(\hat{x}_k) + Q$$

(2)

$$K(t_k) = P(t_{k+1})H^T(\hat{x}(t_{k+1}))H(\hat{x}(t_{k+1}))P(t_{k+1})H^T(\hat{x}(t_{k+1}))+R^{-1}$$

(3)

$$\tilde{x}(t_k) = \hat{x}(t_k) + K(t_k)H(\hat{x}(t_{k+1}))P(t_{k+1})$$

(4)

$$H(\hat{x}(t_{k+1})) = \frac{\partial h(x)}{\partial x} \bigg|_{x = \hat{x}(t_{k+1})}$$

(5)

$$F(\hat{x}(t_k)) = \frac{\partial f(x,u,\alpha^A(t_k))}{\partial x} \bigg|_{x = \hat{x}(t_k)}$$

(6)

where $\hat{x}(t)$ means that it is a predicted value at the $t_{k+1}$ instant and it is calculated on the measurements up to the $t_k$ instant; $\hat{x}(t_{k+1})$ is a corrected state vector estimation; $\hat{x}(t_{k+1})^\text{ex}$ means the extrapolation of a state vector; $P(t_{k+1})$ – a covariance matrix of an extrapolation error, $\hat{x}(t_k)$ – a covariance matrix of an estimation error, $F(\hat{x}(t_k))$ – a gradient.
matrix, \( Q \) – a covariance matrix of a system noise, \( R \) – a covariance matrix of the measurement noise, \( K(t_k) \) – the EKF gain matrix, \( H(\hat{x}(t_{k+1}|k)) \) – a sensitivity matrix,
\[
h(\hat{x}(t_{k+1}|k)) = \\
\begin{bmatrix}
\hat{i}_m(t_{k+1}|k) \\
\hat{i}_d(t_{k+1}|k) \\
\hat{\psi}_m(t_{k+1}|k) \\
\hat{\psi}_d(t_{k+1}|k)
\end{bmatrix}
\] – the result of multiplying the output matrix by a state vector \( \hat{x}(t_{k+1}|k) \); \( f(\hat{x}(t_k), \hat{u}_s^{\alpha \beta}(t_k)) \) – the amount of the state matrix, multiplied by a state vector \( \hat{x}(t_k) \), and an entry matrix, multiplied by a control input vector \( \hat{u}_s^{\alpha \beta}(t_k) \); \( T = (t_k - t_{k-1}) \) – a sampling time.

The estimation of the rotor speed \( \hat{\omega}_r(t_k) \) is calculated in the equation (4).

5 Estimation results

The difference equations of the extended Kalman filter were used to write the MEX function. The parameters of the induction motor used in the EKF mathematical model are shown in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>37.3 kW</td>
</tr>
<tr>
<td>Poles</td>
<td>4</td>
</tr>
<tr>
<td>Voltage</td>
<td>460 V</td>
</tr>
<tr>
<td>Motor speed</td>
<td>1800 rpm</td>
</tr>
<tr>
<td>Frequency</td>
<td>60 Hz</td>
</tr>
<tr>
<td>( R_s )</td>
<td>0.087 ( \Omega )</td>
</tr>
<tr>
<td>( R_r )</td>
<td>0.228 ( \Omega )</td>
</tr>
<tr>
<td>( L_s )</td>
<td>0.0355 ( H )</td>
</tr>
<tr>
<td>( L_r )</td>
<td>0.0355 ( H )</td>
</tr>
<tr>
<td>( L_{m} )</td>
<td>0.0347 ( H )</td>
</tr>
</tbody>
</table>

The following operation mode of the induction motor was simulated in the Simulink. The induction motor was accelerated to a speed \( \omega_{\text{ref}} = 120 \text{ rad/s} \) and at the time \( t = 2 \text{ sec} \) a load of 100 Nm was applied to the motor. Figure 4 shows the calculated motor speed from the standard MATLAB model – \( \omega_r \) and the speed estimation \( \hat{\omega}_r \) – obtained with the extended Kalman filter.

The designed and tested program of the extended Kalman filter can be further used to design a microprocessor-based sensorless control system for an induction motor [10]-[15].

Fig. 4. \( \omega_r \) – an angular rotor speed of the induction motor; \( \hat{\omega}_r \) – an estimation of the rotor speed by the extended Kalman filter.

6 Conclusions

The paper proposed the software and hardware solution, which enables to test different algorithms of a data processing on the basis of the evaluation board STM32F4DISCOVERY. The considered modeling method makes it possible to use the advantages of such programming languages as C, C#, MATLAB in the modeling of electronic device mathematical models.

References


