An Approach to Implementing Functional Blocks of the PLCopen Specification

N. D. Baikov[1], A. N. Godunov[2]

[1]Scientific Research Institute for System Analysis of the National Research Centre Kurchatov Institute, Moscow, Russia, nknikita@niisi.ras.ru;

[2]Scientific Research Institute for System Analysis of the National Research Centre Kurchatov Institute, Moscow, Russia, nkag@niisi.ras.ru;

Abstract. This paper examines possible approaches to implementing functional blocks for a control library for electrical power drives that is being developed as a replacement for foreign software in automated process control systems.

Keywords: function block, motion control library, process control system, PLC

1. Introduction

This article is an authorized translation of the article in Russian, published in this issue of the journal.

Today, the widespread adoption of new process control systems, as well as the maintenance and improvement of the existing ones, is a priority for the economy. Due to its high importance and the diversity of application areas that need specific process control systems, over the years, many suppliers of equipment and off-the-shelf solutions have emerged on the market. Industry standards and other regulatory documents promote equipment compatibility. The most significant ones are IEC 61800 [1], IEC 61131 [2], and PLCopen's [3] specification.

The IEC 61800 standard consists of several parts representing requirements for performance and other aspects of adjustable speed electric drives. In particular, IEC 61800 includes several parts from IEC 61800-7-*, which specify communication and function profiles for adjustable speed electrical power drive systems. These requirements are based on the CiA 402 [4] specification, which defines commands for switching the electric drive modes. Some of these modes are moving the drive to a specified position or accelerating it to a specified velocity. Commands from the CPU module of the programmable logic controller (PLC) to the electric drive can be delivered in a number of ways, such as via the EtherCAT, PROFINET, EtherNet/IP, POWERLINK, Modbus TCP, and CANopen protocols standardized in IEC 61800.

As a rule, application software vendors do not use the operating modes listed in the IEC 61800 standard to describe the electric drive control logic. Application software is created in integrated development environments that provide a higher level of abstraction. The main components in such environments are function blocks (FBs) with their input and output variables. Each FB, when executed, makes the electric drive perform a specific operation. FBs can define more complex modes and operations compared to those defined in the IEC 61800 standard. For example, there is the MC_PositionProfile FB. It sequentially moves the drive to a number of positions at specified intervals. The application software developer combines different FBs to obtain the desired electric drive behavior. For this purpose, there are graphic programming languages, such as ST (structured text language) or the FBD (function block diagrams). The syntax and semantics of these and other languages are standardized in one of the parts of the IEC 61131-3 standard. The PLCopen specification defines the range and interfaces of the recommended FBs (it is supported by many popular development environments, such as CODESYS or Beckhoff TwinCAT). In this way, a development environment translates FB instances into electric drive control commands in accordance with the IEC 61800 standard. For this, motion control libraries and code generation tools are included in the integrated development environment.

Most of these process control system development environments are proprietary and belong to international vendors. There is a demand for similar domestic solutions.

In this study, we analyzed several algorithms and presented a possible implementation of the MC_PositionProfile FB. We used an application runtime environment for Baget-PLC1/2 [5, 6] running on the Baget 3.x real-time operating system [7], and open-source development environment Beremiz [8].

2. MC_PositionProfile FB Interface

MC_PositionProfile is one of the FBs recommended for implementation in the PLCopen specification. According to the specification, one of the required input variables is TimePosition. It is a pointer to an array of pairs (dT_i, p_i) of arbitrary length N. For simplicity, we excluded buffering when multiple FB instances are executed in a sequence (this is controlled by the BufferMode input variable, which is optional according to the PLCopen specification). We only considered an immediate drive start upon executing an instance of the MC_PositionProfile FB. The electric drive must then move sequentially from its current position p_0 to positions $p_i, i = \overline{1, N}$ at specified intervals $dT_i, i = \overline{1, N}$. The PLCopen specification does not explicitly define the trajectory connecting the positions $p_i, i = \overline{1, N}$. There are many ways to solve this problem. We will discuss some of them in the following sections.

3. PV Electric Drive Mode

We used the Profile Velocity Mode (PV) to find possible trajectories for the MC_PositionProfile FB. Here is a brief description of this mode. According to the CiA 402 specification incorporated in the IEC 61800 standard, when the PV mode is on, the electric drive moves according to the velocity curve shown in Fig. 1.

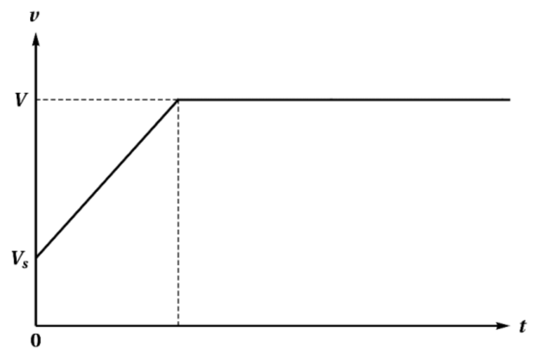

Fig. 1. PV mode velocity vs. time curve

The control parameters in PV mode are:

-V is the drive velocity to be reached and then maintained

-a is the drive acceleration (a positive value)

-d is the drive deceleration (also a positive value).

Fig. 1 shows that the drive velocity V_s at the initial moment may be non-zero (for example, if PV mode activation command interrupts the execution of a previously received command).

Also note that for the drive movement shown in Fig. 1 both velocities V_s and V have the same sign. There are two segments: velocity gain with acceleration a (since $0 < V_s < V$), and movement at constant velocity V. If the initial and final velocities have different signs, and a and d are not equal, the velocity curve consists of three sections because when the curve crosses the zero line, the slope angle changes (the drive goes from deceleration d to acceleration V_s). Below, we considered only the $a = d$ case. For this case, the velocity curve consists of two sections even if the signs of $V_s$and V differ (the acceleration and deceleration slope angles are equal).

4. Option 1. Piecewise Linear Velocity Function with Constant Velocity Segments

As stated above, multiple trajectories pass through positions p_i at intervals dT_i. For example, there is a family of solutions where the electric drive comes to a complete stop at each intermediate position p_i. However, most such solutions are impractical as they include frequently alternating acceleration and deceleration segments to ensure a complete stop at positions p_i. Let us find another solution with a smoother velocity variation when moving through positions p_i.

To make the problem unambiguous, we introduce a control coefficient $0 < \alpha < 0.5$. We seek a solution that meets the condition:

- in the $\sum_{k=1}^{i-1} dT_k + \alpha dT_i \le t \le \sum_{k=1}^{i-1} dT_k + (1-\alpha)dT_i$, $i = \overline{1, N}$ sections, the drive moves steadily at constant velocities V_i

- on the remaining sections, the drive velocity is a linear function of time t. It must be continuous to switch between $V_i$and V_{i+1}.

Boundary conditions should also be met. At the initial moment $t = 0$, the drive must be at position p_0 and velocity V_s. The boundary condition at the final position p_N is $V = V_e$, where V_e is the desired final velocity. In the simplest case, it can be zero, and the drive stops as the MC_PositionProfile FB execution is completed.

The velocity curve for the sought solution is shown in Fig. 2.

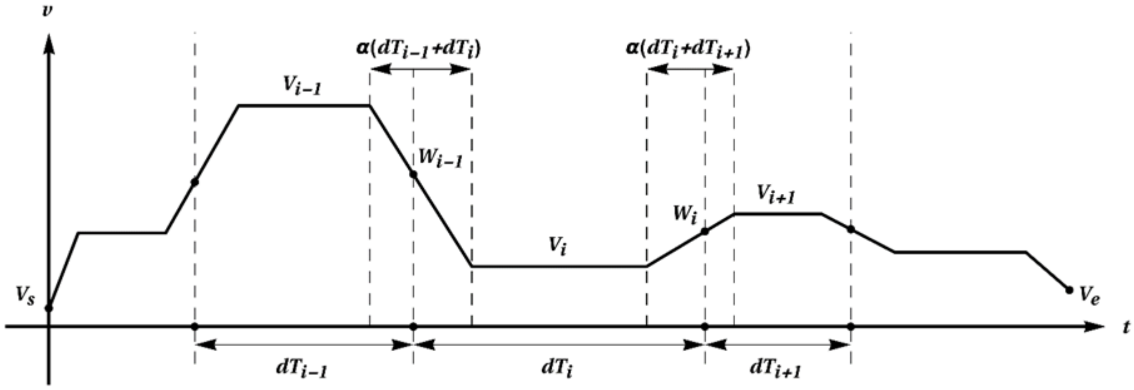

Fig. 2. Velocity curve for the sought solution

Let us denote the velocities at points $\sum_{k=1}^{i} dT_k\,, i = \overline{0, N}$ as W_i ($i = 0$ is at point $t = 0$). Since the desired velocity function is linear, we obtain:

$$W_i = V_i + \frac{V_{i+1} - V_i}{\alpha(dT_i + dT_{i+1})}\alpha dT_i = \frac{dT_i}{dT_i + dT_{i+1}}V_{i+1} + \frac{dT_{i+1}}{dT_i + dT_{i+1}}V_i,$$
$$i = \overline{1, N-1}$$
$$W_0 = V_s, \qquad W_N = V_e$$

The drive reaches velocities W_i when reaching positions p_i.

Let us calculate the coefficients of the system of linear equations to find the elements of the velocity array $\{V_i,\ i = \overline{1, N}\}$. Note that the change in position at each time interval depends on the area under the velocity curve:

$$p_i = p_{i-1} + \frac{W_{i-1} + V_i}{2}(\alpha dT_i) + V_i(1 - 2\alpha)dT_i + \frac{V_i + W_i}{2}(\alpha dT_i), \quad i = \overline{1, N}$$

Therefore, we obtain

$$2\frac{p_i - p_{i-1}}{dT_i} = (W_{i-1} + V_i)\alpha + 2V_i(1 - 2\alpha) + (V_i + W_i)\alpha, \quad i = \overline{1, N}$$

For $i = \overline{2, N-1}$, we substitute W_{i-1} and W_i in the equation above with V_{i-1}, V_i and V_{i+1}m, and combine like terms:

$$2\frac{p_i - p_{i-1}}{dT_i} = \frac{\alpha dT_i}{dT_{i-1} + dT_i}V_{i-1} + \left(2 - 2\alpha + \frac{\alpha dT_{i-1}}{dT_{i-1} + dT_i} + \frac{\alpha dT_{i+1}}{dT_{i+1} + dT_i}\right)V_i + \frac{\alpha dT_i}{dT_{i+1} + dT_i}V_{i+1}, \qquad i = \overline{2, N-1}$$

To deduct equations for $i = 1$ and $i = N$, we note that $W_0 = V_s$ and $W_N = V_e$:

$$2\frac{p_1 - p_0}{dT_1} - \alpha V_s = \left(2 - 2\alpha + \frac{\alpha dT_2}{dT_1 + dT_2}\right)V_1 + \frac{\alpha dT_1}{dT_1 + dT_2}V_2,$$

$$2\frac{p_N - p_{N-1}}{dT_N} - \alpha V_e = \frac{\alpha dT_N}{dT_{N-1} + dT_N}V_{N-1}\left(2 - 2\alpha + \frac{\alpha dT_{N-1}}{dT_{N-1} + dT_N}\right)V_N$$

By now, we have obtained a system of linear equations with a tridiagonal matrix of coefficients that returns unknown velocities V_i, $i = \overline{1, N}$. The matrix is strictly diagonally dominant (for each element of the main diagonal, its absolute value exceeds the sum of the absolute values of all other elements in the same row). We can apply the tridiagonal matrix algorithm to solve such a system of linear equations [9].

Accelerations can be found from the values of the velocity function V_i found through solving the system of linear equations:

$$a_0 = \frac{V_1 - V_s}{\alpha\ dT_1}$$
$$a_i = \frac{V_{i+1} - V_i}{\alpha(dT_i + dT_{i+1})}, \quad i = \overline{1, N-1}$$
$$a_N = \frac{V_e - V_N}{\alpha\ dT_N}$$

The final MC_PositionProfile control function logic is as follows:

- at the initial moment $t = 0$, the electric drive receives a command to switch to the PV mode with the control parameters $V = V_1$ and $a = d = |a_0|$
- at each moment $\sum_{k=1}^{i-1} dT_k + (1 - \alpha)dT_i,\ i = \overline{1, N-1}$, a new command is sent to the drive to switch to the PV mode with the control parameters $V = V_{i+1}$ and $a = d = |a_i|$
- at the moment $\sum_{k=1}^{N-1} dT_k + (1 - \alpha)dT_N$, the last command to switch to the PV mode is sent to the drive with control parameters $V = V_e$ and $a = d = |a_N|$.

5. Option 2. Cubic Spline

An alternative implementation of the MC_PositionProfile FB uses a cubic spline to represent the drive trajectory. We assume that the CPU module of the drive PLC can send commands to switch to the PV mode with updated control parameters V, a, and d at a sufficiently high frequency ν.

To deduce equations for the cubic spline coefficients, let us introduce some notation. Let

$$t_0 = 0, \qquad t_i = \sum_{k=1}^{i} dT_k, \qquad i = \overline{1,N}$$

Let us approximate the drive position at each of the intervals $[t_i, t_{i+1}], i = \overline{0, N-1}$ using the following functions:

$$P_i(t) \coloneqq A_i + B_i(t - t_i) + C_i(t - t_i)^2 + D_i(t - t_i)^3$$

The cubic spline must meet the following conditions:

$$P_0(t_0) = p_0,$$
$$P_{i-1}(t_i) = P_i(t_i) = p_i, \qquad i = \overline{1, N-1},$$
$$P_{N-1}(t_N) = p_N$$
$$P_i'(t_{i+1}) = P_{i+1}'(t_{i+1}), \qquad i = \overline{0, N-2}$$
$$P_i''(t_{i+1}) = P_{i+1}''(t_{i+1}), \qquad i = \overline{0, N-2}$$

Let us denote two auxiliary coefficients $A_N \coloneqq P_{N-1}(t_N)$ and $C_N \coloneqq P_{N-1}''(t_N)/2$.

We reduce the problem of finding the cubic spline coefficients to solving a system of linear equations in the unknowns C_i. For this, we express all remaining unknowns A_i, B_i, D_i in terms of C_i.

It follows from the conditions

$$P_i(t_i) = p_i, \qquad i = \overline{0, N-1}$$
$$A_N = P_{N-1}(t_N)$$

that

$$A_i = p_i, \qquad i = \overline{0, N}$$

From the condition

$$P_i''(t_{i+1}) = P_{i+1}''(t_{i+1}), \qquad i = \overline{0, N-2}$$

we obtain

$$D_i = \frac{C_{i+1} - C_i}{3\, dT_{i+1}}, \qquad i = \overline{0, N-2}$$

From the condition

$$P_i(t_{i+1}) = P_{i+1}(t_{i+1}), \qquad i = \overline{0, N-2}$$

we obtain

$$B_i = \frac{A_{i+1} - A_i}{dT_{i+1}} - \frac{dT_{i+1}}{3}(C_{i+1} + 2\, C_i), \quad i = \overline{0, N-2}$$

Let us substitute the expressions for D_i and B_i into the condition

$$P_i'(t_{i+1}) = P_{i+1}'(t_{i+1}), \qquad i = \overline{0, N-2}$$

and combine the terms with identical C_i. The result is the following system of linear equations:

$$\frac{dT_{i+1}}{3}\, C_i + \frac{2}{3}(dT_{i+1} + dT_{i+2})\, C_{i+1} + \frac{dT_{i+2}}{3}\, C_{i+2} = \frac{A_{i+2} - A_{i+1}}{dT_{i+2}} - \frac{A_{i+1} - A_i}{dT_{i+1}}, \quad i = \overline{0, N-2}$$

To achieve system closure, two more linear equations must be added. To obtain them, we use the boundary conditions: the drive velocity at the initial moment must be V_s, and at the final moment, V_e. In this case, from the condition

$$V_s = P_0(t_0) = B_0 = \frac{A_1 - A_0}{dT_1} - \frac{dT_1}{3}(C_1 + 2\, C_0)$$

we obtain an additional equation

$$\frac{2}{3} dT_1\, C_0 + \frac{1}{3} dT_1\, C_1 = \frac{A_1 - A_0}{dT_1} - V_s$$

Next, we obtain a similar equation from V_e. It follows from the conditions

$$P_{N-1}(t_N) = A_N$$
$$P_{N-1}'(t_N) = V_e$$

that

$$A_N - dT_N\, V_e = P_{N-1}(t_N) - dT_N P_{N-1}'(t_N) = A_{N-1} - C_{N-1} dT_N^2 - 2 D_{N-1} dT_N^3$$

Let us express D_{N-1} from the identity:

$$D_{N-1} = -\frac{A_N - A_{N-1}}{2\, dT_N^3} + \frac{V_e}{2 dT_N^2} - \frac{C_{N-1}}{2\, dT_N}$$

Let us substitute D_{N-1} into the condition

$$C_N = \frac{P_{N-1}''(t_N)}{2} = C_{N-1} + 3\, dT_N\, D_{N-1}$$

As a result, we obtain an additional equation

$$\frac{1}{3} dT_N\, C_{N-1} + \frac{2}{3} dT_N\, C_N = -\left(\frac{A_N - A_{N-1}}{dT_N} - V_e\right)$$

Once again, we obtained a system of linear equations with a tridiagonal matrix that can be used to find unknown coefficients C_i, $i = \overline{0, N}$. It can also be solved with the tridiagonal matrix algorithm. To obtain the spline function, we substitute the system solutions C_i into the expressions for the coefficients A_i, B_i, D_i presented in the equations above.

At each interval $[t_i, t_{i+1}], i = \overline{0, N-1}$ the velocity$V$ passed as a parameter in the PV mode activation command is calculated every time from the spline function as $V = P_i'(t + 1/\nu)$, where t is the current time, and ν is the command sending frequency. The parameters a and d are calculated as $a = d = |(V - V_{cur})\, \nu|$, where V_{cur} is the current drive velocity at the moment of sending a new command.

The cubic spline MC_PositionProfile FB algorithm is as follows:

- Upon calling an FB MC_PositionProfile instance, the coefficients of the spline A_i, B_i, C_i, D_i are calculated
- A command to switch the drive to the PV mode with new control parameter values V, a, and d calculated from the equations above is sent to the drive at the frequency specified in the CPU module settings

It should be noted that, unlike the first option, the cubic spline algorithm applies to reach the positions p_i, an error occurs because the drive velocity does not follow the spline but changes in steps. Therefore, before applying Option 2, we must estimate the error that depends on the frequency ν and select such a frequency ν that the resulting error does not negatively affect the controlled manufacturing process.

6. Conclusion

This study examined possible implementations of FBs compliant with the PLCopen specification for the Baget-PLC1/2 application runtime environment running on the Baget 3.x real-time operating system.

This study is a part of the Deployment of Trusted AI Systems based on New Mathematical and Algorithmic Approaches and Fast Computing Models Compatible with Domestic Computer Hardware, a government contract for fundamental research (FNEF-2024-0001 stage 2025, 1023032100070-3-1.2.

References

1. IEC 61800:2021 Adjustable Speed Electrical Power Drive Systems. International Electrotechnical Commission. 2021.
2. IEC 61131:2023 Programmable Controllers. International Electrotechnical Commission. 2023.
3. PLCopen Technical Committee 2. PLCopen Motion Control – Function Blocks – Part 1 & 2. Version 2.0. PLCopen. 2021.
4. CiA 402:2024 Device Profile for Drives and Motion Control. CAN in Automation. 2024.
5. BAGET-PLC1 Operating Manual YKSU.421457.002RE. Available at: https://niisi.ru/БАГЕТ-ПЛК1_РЭ.pdf (last accessed: October 16, 2025) (in Russ.)
6. BAGET-PLC2 Operating Manual YKSU.421457.003RE. Available at: https://niisi.ru/РЭ_ПЛК2_изм1.pdf (last accessed: October 16, 2025) (in Russ.)
7. A.N. Godunov, V.A. Soldatov. Operating Systems Of The Baget Family (Likeness, Differences and Perspectives) Programming and Computer Software, (2014), No. 5, pp. 68–76.
8. Beremiz project. Available at: https://beremiz.org (last accessed: October 16, 2025)
9. N.S. Bakhvalov, N.P. Zhidkov, G.M. Kobelkov. Numerical Methods. Moscow, Knowledge Laboratory, 2023 (in Russ.)