

Article

The Impact of Computational Accuracy on the Quality of Direct Drive Control

Grzegorz Góra ¹, Maciej Petko ¹, Konrad Gac ¹, Jakub Górski ¹, Joanna Iwaniec ^{1,*}, Michał Mańka ¹
and Wojciech Zabierowski ^{2,*}

¹ AGH University of Krakow, Faculty of Mechanical Engineering and Robotics, Department of Robotics and Mechatronics, Al. Mickiewicza 30, 30-059 Krakow, Poland; ggora@agh.edu.pl (G.G.); petko@agh.edu.pl (M.P.); kgac@agh.edu.pl (K.G.); jgorski@agh.edu.pl (J.G.); mmanka@agh.edu.pl (M.M.)

² Department of Microelectronics and Computer Science, Lodz University of Technology, ul. Wólczajska 221, 93-005 Łódź, Poland

* Correspondence: jiwaniec@agh.edu.pl (J.I.); wojciech.zabierowski@p.lodz.pl (W.Z.)

Abstract: One of the main aspects of the control system development process for direct drives is the selection of the required computational accuracy while establishing its impact on the quality of the control. Understanding this relationship allows designers to consciously determine the system structure at the early stages of controller development, which enables the optimal usage of hardware resources. This paper analyzes the results of experimental research on the influence of computational accuracy on the quality of control of direct drives. During the carried-out research, several vector controllers with different computational precision were implemented using Field-Programmable Gate Arrays (FPGAs). The experiments were carried out on a dedicated research stand for testing direct drives. Test scenarios included position and trajectory monitoring under various torque loads. To assess the control quality, the measures based on the deviations from the value set by the controller were proposed. In this paper, the results of experiments have been presented in the form of values of the measures in relation to the computational accuracy. The obtained results proved that satisfactory drive operation parameters can be obtained despite the relatively low accuracy of calculations in the control algorithm.

Keywords: direct drives; influence of computational accuracy on the quality of control of direct drives; field-programmable gate arrays (FPGA); controller development



Citation: Góra, G.; Petko, M.; Gac, K.; Górski, J.; Iwaniec, J.; Mańka, M.; Zabierowski, W. The Impact of Computational Accuracy on the Quality of Direct Drive Control. *Electronics* **2024**, *13*, 1052. <https://doi.org/10.3390/electronics13061052>

Received: 15 January 2024

Revised: 27 February 2024

Accepted: 8 March 2024

Published: 12 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Direct drives constitute an important group of electric drives. Due to their structure, they allow for high torque at low speeds, which enables the removal of the gear from the kinematic chain. As a result, the control accuracy and reliability of the system are higher, costs are reduced, and drives can be integrated into the structure [1]. Hence, direct drives find application across different industries, such as electric vehicles, industrial and service robotics, and the metallurgical sector [2]. The percentage share of direct drives in those applications is expected to increase due to the requirement to reduce greenhouse gas emissions.

The design of most direct drives is based on the construction of brushless direct current (BLDC) [3] and permanent magnet synchronous motors (PMSMs) [4]. A common feature of these motor types is the usage of permanent magnets in the rotor and windings in the stator, generating a rotating magnetic field. However, the main difference is the shape of the back electromotive force (BEMF) as a function of the rotor rotation angle. In the case of a BLDC motor, it has a shape similar to a trapezoidal function, while in the case of a PMSM, it has the shape of a sinusoidal function [5,6]. The BLDC motor generates a higher peak torque compared to PMSM, which, in turn, is associated with a higher level of ripples that results in the generated noise and worse performance at low rotational speeds [5]. These

properties determine the choice of control algorithms, which, for BLDC and PMSM, can be classified into two categories: scalar and vector.

In the scalar techniques, the speed or torque of the motor is controlled without precise determination of the rotor position. This position is roughly estimated based on the readings from the Hall effect sensors [7,8] or via BEMF [9,10]. On this basis, the controller turns on the bridge transistors in the appropriate order, forcing the rotor to rotate. The speed and torque values are controlled by adjusting the voltage supplying the motor [11,12] or by changing the pulse width (PWM) controlling the bridge transistors [7,10,13]. The implementation of scalar methods is relatively simple and does not require complex measurement systems. However, they do not provide accurate control of the rotor position, which may be significant for more demanding industrial applications.

The second group of control techniques are vector methods. In these methods, the stator winding currents are identified as two orthogonal components that can be represented by a vector. One component determines the magnetic flux, and the other determines the torque. On this basis, the control system calculates the appropriate current components of the stator windings, which causes the rotor to rotate with the required speed and torque. In the case of the vector techniques, knowledge concerning the rotor position is required. In practical applications, it is determined based on the data from the encoder [14,15] or estimated based on the model and measured values of the winding currents [16].

The vector approach allows for the development of different versions of the algorithms, which can be classified into the following groups [17,18]: field-oriented control (FOC) [16,19,20] and direct torque control (DTC) [21,22]. In FOC, the control algorithm transforms the measured values of currents and rotor positions into a system of two coordinates related to the rotor. This vector is passed to the controller to calculate the control voltage vector. In the case of DTC, the flux and torque variables are used to select voltage vectors. These data are processed by the drive system controller to analyze/determine the commutation point.

Vector techniques require much more complex computations than the scalar methods. The requirement to use complex logic circuits increases the cost of the controller significantly. However, from year to year, the prices of these circuits are becoming lower. In the near future, the vector methods may become the standard methods for controlling BLDC and PMSM motors [23].

Due to the mentioned requirements, various logic circuits are used to implement vector control algorithms. One of the main groups of these systems is represented by digital signal processors (DSPs) [19]. The structure of these systems includes many dedicated blocks, such as analog-to-digital and digital-to-analog converters, as well as systems for quickly performing operations on digital values. Such architecture allows DSP systems to perform complex calculations in a short time.

The second group of logical systems on which vector controllers are based are Field-Programmable Gate Arrays (FPGAs) [24,25]. These systems contain a matrix of logical elements (LEs), registers, multipliers, and input/output interfaces. The connections between subsystems can be freely configured to develop a custom system, which facilitates parallel data processing and shortens the time required to perform critical procedures. The mentioned features make FPGAs more flexible compared to DSPs. However, the system design requires simultaneous planning of the structure of the logical system and the software [26].

Due to the ease of configuration, it is possible to use the FPGA systems with fewer LEs instead of complex DSPs. Moreover, the FPGAs with more resources can sustain the circuit to control multiple engines simultaneously in a single chip. However, to develop such a controller, it is necessary to know what computational accuracy level is sufficient for efficient vector control. Such knowledge makes it possible to design systems with optimal usage of FPGA resources in terms of occupancy, which is the issue discussed in this paper. The main scientific contributions of this paper are as follows:

- Research the influence of computational accuracy on the quality of FOC control. The impact of different levels of computational accuracy on motor position and torque control has been investigated;
- Analysis of the impact of computational accuracy on the usage of FPGA resources.

The manuscript is structured as follows. Section 2 concerns the field-oriented control (FOC) method. Section 3 presents the test stand dedicated to the experimental investigation of direct drive performance. Section 4 contains an experimental research description related to tracking values and trajectories. Section 5 provides the analysis of the obtained experimental results. A discussion of the obtained results is presented in Section 6. Finally, this paper is summarized and concluded in Section 7.

2. Field-Oriented Control

The field-oriented control (FOC) method was developed for induction motors. This algorithm transforms the description of the three-phase motor current and voltage relationships into equations in a two-dimensional coordinate system to obtain the decoupling of magnetic flux and torque, which occurs naturally in DC motors [27,28]. In induction motors and permanent magnet drives, independent flux, and torque control can be achieved by controlling the current vector components in the coordinate system associated with the rotor [29]. This assumption is the basis of the field-oriented vector control method. The transformation between a three-phase system and a system rotating with the angular speed of the rotor is performed with the application of the Clarke and Park transformations (Figure 1).

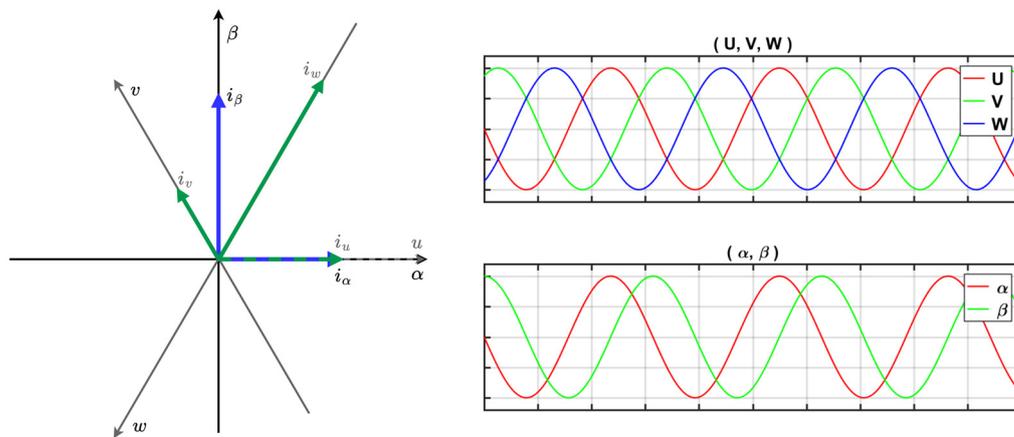


Figure 1. Clarke transformation from 3-dimensional space (u, v, w) to 2-dimensional space (α, β). Components $i_u, i_v,$ and i_w represent currents of 3 motor phases, while i_α and i_β represent orthogonal current components, which result from the Clarke transformation.

The Clarke transformation is used to transform the resultant stator current vector, which has a composition rotating in a three-dimensional phase space (u, v, w), into a two-dimensional stationary orthogonal coordinate system (α, β) associated with the stator [30]. The α axis coincides with the axis of the first phase u , while the β axis is perpendicular to it (Figure 1). Thanks to the Clarke transformation, the current vector can be represented by two components instead of three phase components, according to the following equations:

$$\begin{aligned}
 I_\alpha &= I_U, \\
 I_\beta &= \frac{1}{\sqrt{3}}I_U + \frac{2}{\sqrt{3}}I_V.
 \end{aligned}
 \tag{1}$$

The geometric sum of currents in the motor is always equal to zero. Therefore, measuring two-phase currents is sufficient to obtain complete information about the position and amplitude of the current vector in the space. The inverse Clarke transforms two-phase quantities (α, β) into three-phase quantities (u, v, w).

The Park transformation is used to transform vector quantities described in the system (α, β) into the orthogonal coordinate system (d, q) , rotating synchronously with the angular velocity of the rotor θ_e (Figure 2). The transformation to the (d, q) coordinate system eliminates time-varying inductances from the voltage equations of a three-phase drive. The parameter appearing in the transformation equations is the instantaneous angular position of the rotor:

$$\begin{aligned} I_d &= I_\alpha \cos(\theta_e) + I_\beta \sin(\theta_e), \\ I_q &= -I_\alpha \sin(\theta_e) + I_\beta \cos(\theta_e). \end{aligned} \quad (2)$$

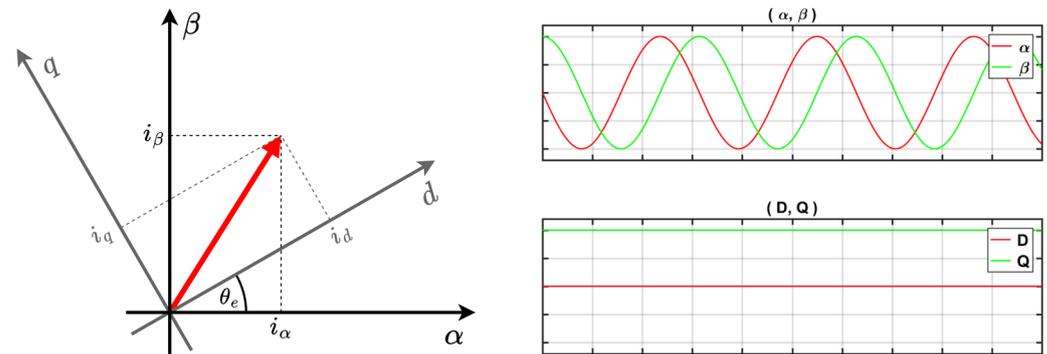


Figure 2. Park transformation from the stationary coordinate system (α, β) to the rotating coordinate system (d, q) . The components i_α and i_β represent the orthogonal components of the stator current, the results of the Clarke transformation in a stationary system of coordinates. The components i_d and i_q (longitudinal and transverse currents in the magnetic axes) describe the same values in a rotating system of coordinates, according to the function of the angle ϕ .

As a result of the Park transformation application, two DC components are obtained, which have their own physical interpretation. The current component i_d allows for a change in the value of the magnetic flux in the direction of the d -axis. The i_q component is interpreted as a value proportional to the motor torque [31]. In order to achieve the highest possible engine efficiency, the commutation algorithm should minimize the value of the i_d component and maximize the value of the i_q component. The Park transformation is also associated with its inverse transformation, which transforms quantities from a rotating coordinate system (d, q) to a stationary system (α, β) .

The basic advantages of the FOC algorithm are high torque control dynamics, high efficiency in a wide speed range, and precise torque control [32]. The diagram of the FOC control method is shown in Figure 3. Commutation is achieved by performing several steps. The values of phase currents (I_u, I_v) are measured and converted into longitudinal and transverse currents (I_q, I_d) using Clarke and Park transformations and rotor position θ_m . The calculated components are subtracted from the reference values (i_{qref}, i_{dref}) to calculate the error for proportional–integral (PI) controllers, which return the input data (u_{qref}, u_{dref}), which is transformed by the inverse Park transform ($u_{\alpha ref}, u_{\beta ref}$) to control the power stage.

To fully utilize the electrical power, the motors should operate at the maximum ratio of the electromagnetic torque to the supply current T_e/I . Thus, the Clarke–Park transformation is used, converting the stator phase inductances L_u, L_v , and L_w into longitudinal inductances L_d and transverse inductances L_q in the magnetic axes. If the inductances are equal ($L_d = L_q$), the maximum ratio of torque to current is achieved for the phase angle between the stator current vector and the magnetic flux vector of permanent magnets equal to 90° . Such a situation is equivalent to powering the motor winding with the so-called transverse current only, i.e., $I = I_q$. Inductance symmetry is characteristic of motors with magnets mounted on the SPM (surface permanent magnet) rotor surface.

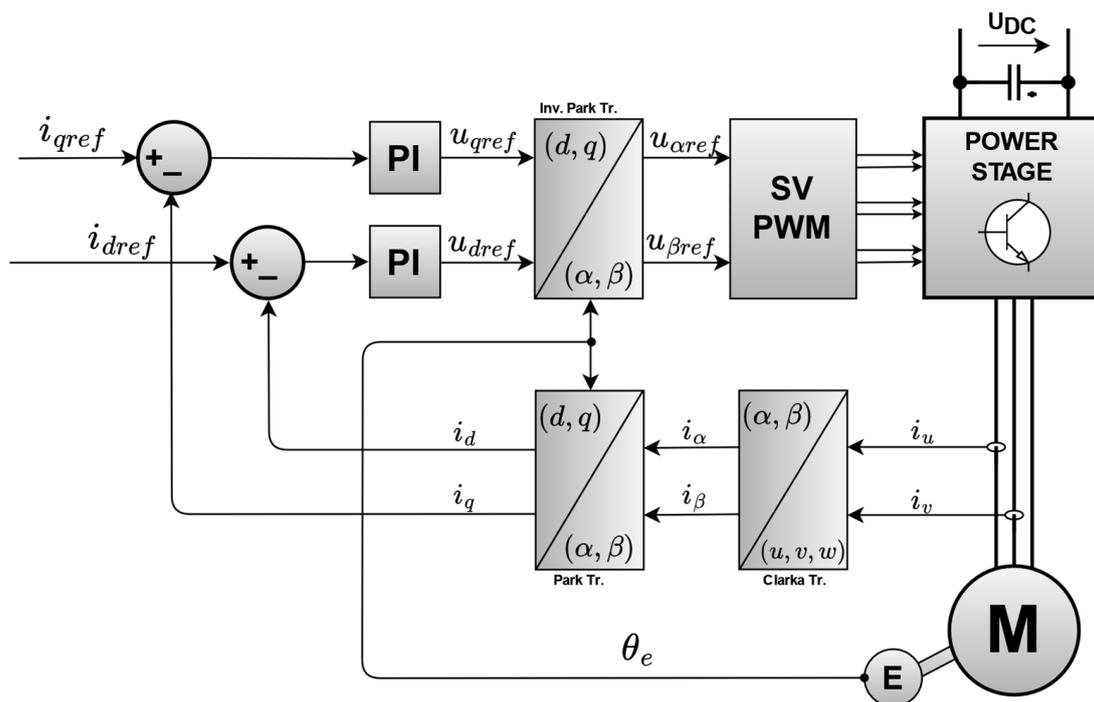


Figure 3. Diagram of the FOC method.

In the case of a motor for which the transverse inductance is greater than the longitudinal inductance ($L_d < L_q$), the optimum T_e/I ratio occurs when the supply current vector leads the rotor magnetic flux vector by a phase angle greater than 90° . Magnetic asymmetry is often encountered in motors with magnets mounted inside the rotor core (interior permanent magnet, IPM). For this type of drive, the i_q component is controlled while maintaining a linear relationship between the current and the torque (keeping $i_d = 0$), or both components are controlled with a non-zero value to produce the maximum torque (maximum torque per ampere, MTPA) [33]. In such a case, the value of the i_d component is determined from the following equation [34], where Ψ_{PM} is permanent magnet (PM) flux linkage.

$$I_d = \frac{\Psi_{PM}}{2 \cdot (L_q - L_d)} - \sqrt{\frac{\Psi_{PM}^2}{2 \cdot (L_q - L_d)^2} + i_q^2} \tag{3}$$

The research presented in the literature aimed at improving control parameters by using complex mathematical models of the MPC (model predictive control) drive [33,35,36] or using controllers based on fuzzy logic [37].

3. Investigation into Electronic Commutation Algorithms

Commercially available direct drive controllers have a closed architecture that makes it impossible to change the controller operating parameters such as, e.g., type of modulation, sampling frequency, computational accuracy, etc. Therefore, it was decided to build an open system that will enable an experimental comparison of the influence of the mentioned operating parameters of the control system on the level of the torque ripples and the control quality of direct drives. For this purpose, a new open system has been developed, the elements of which are as follows:

- Test stand—Built in a modular way, which allows for a quick change of its configuration and easy adaptation to the requirements of the currently conducted research. The test stand enables, among others, the measurement of torque and the application of an external load to the tested drive (inertial or active load through another electric drive);

- Power stage prototypes—Two power stage prototypes have been designed. The first stage is based on the power MOSFET transistors and an integrated driver for MOSFET/IGBT transistors intended for medium-power drives (1.5 kW). The second power stage is based on the bipolar transistors and an optically isolated driver designed for low-power drives (200 W). Both power stages are equipped with circuits allowing for the support of Hall sensors, an incremental encoder, and the measurement of phase currents in two phases of the motor;
- HDL Hardware Module Library—This allows for the implementation of a direct drive control system operating in various modes (position, speed, current) with various levels of complexity and various levels of computational accuracy;
- Application for PC—This enables communication with a hardware platform on which an electronic commutation system (FPGA) has been implemented to write/read operating parameters and record measurement data, additionally allowing online viewing.

3.1. Stand for Testing Electronic Commutation Algorithms

The mechanical part of the test stand is built in a modular manner, which allows for a quick change of its configuration and easy adaptation to the requirements of the currently conducted research (Figure 4). The modularity is achieved by adding, removing, or replacing station components. The elements of the stand are laboratory power supply (PS), torque sensor (TS), electric drive with a controller constituting a load (ML and MLD), electric brake (EB), control panel (CP), power stage of the tested drive (MTI) with its electronic commutation system implemented in the FPGA (Field-Programmable Gate Array) chip (MTC) and the test stand controller (SC).

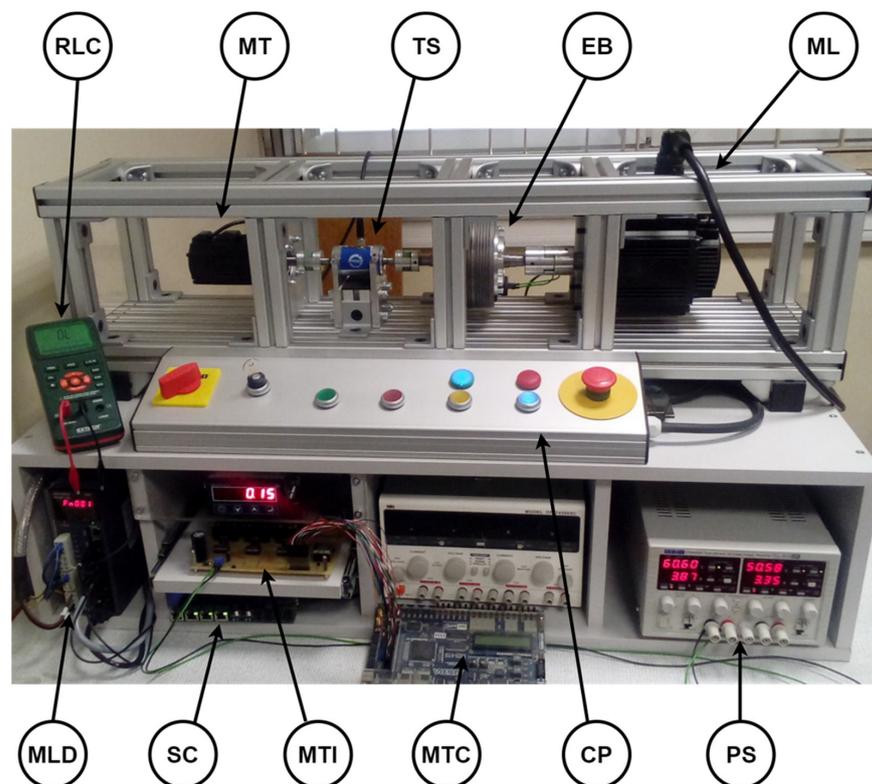


Figure 4. The test stand for testing commutation algorithms of direct drives: RLC—RLC measurement bridge; MT—tested PMSM motor; TS—torque sensor; EB—electric brake; ML—PMSM drive, which states the load for the tested drive; MLD—load drive controller; SC—test stand controller; MTI—tested drive controller power stage; MTC—electronic commutator for MTI power stage implemented in FPGA chip; CP—control panel; PS—laboratory power supply.

The laboratory power supply (PS) is designed to supply the MTI power stage. It has two independent (galvanically isolated) channels, thanks to which it is possible to achieve a DC voltage of up to 120 V or a maximum DC current of up to 40 A. Additionally, the power supply has digital commutation interfaces (RS-232, USB, digital inputs and outputs in TTL voltage standard) for remote control, saving, and reading of operation parameters.

The second element of the stand is a torque sensor (TS), designed to measure the torque transmitted between the tested drive (MT) and the load, which is the ML drive or the electric brake (EB). It generates an analog signal in the range of 0.5–4.5 V, proportional to the load torque. The stand is equipped with three sensors with the following ranges: ± 2.5 Nm, ± 7.5 Nm, and ± 17.5 Nm, which are installed depending on the maximum torque generated by the tested drive (MT). The sensor enables measurement at a rotational speed range from 0 to 5000 rpm, and its bandwidth is 1 kHz. The sensor accuracy declared by the manufacturer is below 0.5%, and the repeatability is below 0.1% of the entire measurement range. Thanks to the application of an independent torque sensor, it is possible to measure the torque generated by the tested motor with high accuracy.

Another element of the test stand is the 1.1 kW PMSM electric motor (ML) with a controller (MLD). The main task of the ML and MLD setup is to generate a static and dynamic torque load for the tested drive and power stage. The maximum continuous torque for this motor is 4.78 Nm (peak torque 14.3 Nm), and the rated speed is 3000 rpm. The system can work in the control loop: torque, speed, and position, so it can be used in a wide range of research applications and experiments.

To generate the load torque for the tested drive, it is also possible to use a power electric brake (EB) built into the stand with braking torque of up to 15 Nm. The given load torque is proportional to the current supplying the brake winding, which allows for smooth adjustment. The brake is also used in static tests and experiments.

The last fixed element of the bench is its controller (SC), which consists of two parts: the Terasic DE0-Nano module with the Intel (Altera) Cyclone IV family FPGA chip and the PCB-printed circuit board, which is an interface with the peripheral systems of the station. The main tasks of the driver shown in Figure 5 include the following:

- Measurement of the analog signal from the torque sensor and its conversion through the built-in 16-bit ADC (analog-to-digital converter);
- Generation of an analog signal in the ± 10 V standard, constituting the load torque setpoint for the (ML) drive controller, through the built-in 16-bit DAC (digital-to-analog converter);
- Communication with the MLD controller through the RS-485 [38] serial interface and digital input and output ports;
- Communication with a PC via a serial port, which enables setting and reading parameters and acquisition of measurement data (e.g., speed, shaft position, signal from a torque sensor);
- Communication with the CP (e.g., changing the operating mode, starting and stopping the measurement, signaling errors and failures);
- Generation of the set value waveforms (position, speed, or torque) and the set load and braking torque through the implemented waveform generator;
- Measurement of the current angular position of the shaft based on signals generated by the MLD controller in the RS-422 standard [39];
- Temperature measurement from four independent channels (e.g., on the power stage, drive, brake).

Additional elements of the test stand are the Extech RLC LCR200 digital meter (Extech Instruments, Nashua, NH, USA), which allows the determination of the resistance and the inductance of the stator windings, and the Agilent Technologies MSO7054B four-channel digital oscilloscope (Agilent Technologies, Inc., Santa Clara, CA, USA).

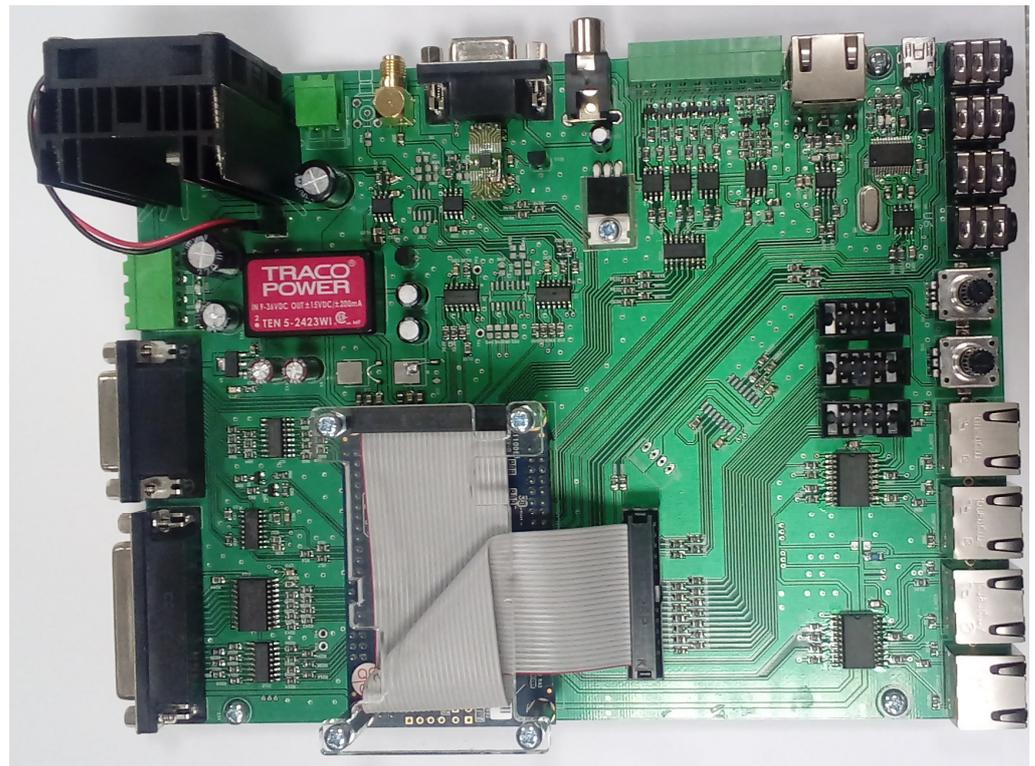


Figure 5. Controller of the stand for testing commutation algorithms of direct drives.

The type and parameters of the generated waveforms are set online before the start of each experiment via the DDSApp application, which cooperates with the station controller. The generator implemented in the station controller makes it possible to determine a trajectory consisting of the following segments:

- Const—constant component: maintaining a constant set value (settable parameters: value, segment duration);
- Ramp—waveform segment described by a linear function (settable parameters: initial value, final value, segment duration);
- Pulse—the sequence of rectangular pulses: trajectory segment intended for testing the step response of the control system (settable parameters: period, positive pulse time, negative pulse time, positive pulse amplitude, negative pulse amplitude, constant component, number of segment repetitions);
- Saw—sawtooth waveform (settable parameters: period, amplitude, constant component, number of segment repetitions);
- Trapeze—trapezoidal waveform (settable parameters: period, amplitude, edge rising time, constant component, number of segment repetitions);
- Sin—a sinusoidal waveform, a smooth waveform with all derivatives continuous (settable parameters: amplitude, period, constant component, number of segment repetitions);
- SinSat—sinusoidal waveform with saturation (cutting): a gentle waveform with a dynamic transition to the static phase maintaining the set value (settable parameters: amplitude, saturation value, period, constant component, number of segment repetitions).

3.2. HDL Hardware Module Library

Direct implementation using hardware description languages (HDL) allows for efficient usage of hardware resources (by performing arithmetic operations on fixed-point numbers with normalized ranges), ensures a constant and high sampling rate, and makes it possible to divide the system into independent units performing tasks in parallel. Additionally, the system designer decides the level of advancement, speed of operation, and the amount of hardware resources used, depending on the predicted requirements.

To conduct experimental research on electronic commutation algorithms for direct drives, a library of hardware modules in the VHDL language was designed, constituting components for the implementation of the drive control system. Basic parameters and assumptions regarding the designed hardware blocks are as follows:

- Base clock frequency: 50 to 100 MHz;
- Position sensor resolution: 8 to 24 bits;
- Operating (sampling) frequency: 2.5 to 40 kHz;
- Maximum phase current of the drive: ± 25 A;
- Maximum drive rotational speed: $\pm 30,000$ rpm;
- Possibility of changing parameters that cannot be set in the commercially available direct drive controllers (e.g., operating frequency, so-called “dead times”);
- Architectural flexibility—the possibility to expand the library with additional elements ensuring cooperation with other types of drives and power stages (e.g., multi-level power stages or other types of PWM modulation).

Figure 6 shows the direct drive control diagram using elements from the Hardware Module Library. The input data for the control system are values of phase currents (i_u, i_v) and the position of the rotor relative to the stationary stator coordinate system θ_m . Then, based on the rotor position, considering the number of pairs of drive poles, the value of the electric angle θ_e is calculated. In the next stage of the control algorithm realization, the current values in the rotor coordinate system (i_d, i_q —after Clarke and Park transformation) are compared with the set values in the double PI controller, which determines the modulation coefficient and the correction angle in relation to the angle θ_e , determining the position of the magnetic flux vector of the permanent magnets. These data, along with the value of the electric angle, constitute the basis for determining the fill factors (S_u, S_v, S_w) for the SVPWM modulator. Then, the signals controlling the keyer elements of the power stage (separately for the lower and upper half-bridge transistors) are generated in the PWM-HL block. The last element of the architecture is a module that allows for communication with the DDSApp application via the UART bus to establish the setting values of the online regulators. The phase current values are read by the phase current measurement module, which communicates with two external ADC converters via the SPI (serial peripheral interface) bus.

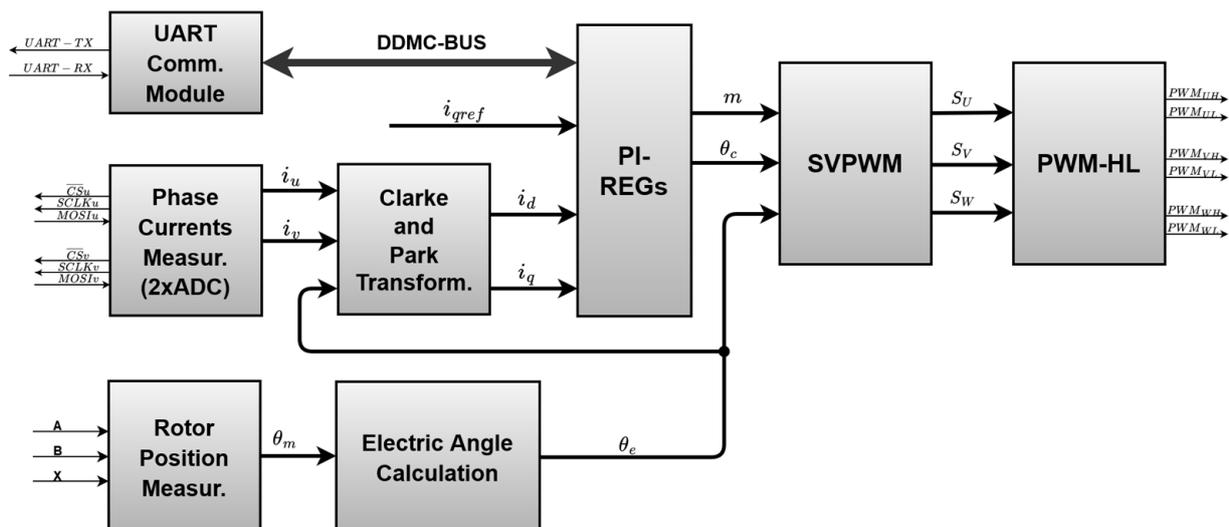


Figure 6. Simplified direct drive control diagram using elements of the Hardware Module Library.

The main task of modules intended for communication with position sensors (rotor position measurement) is to read information about the value of the angle determining the position of the shaft relative to the stationary coordinate system associated with the stator. This value allows for determining the position of the magnetic flux vector coming

from the permanent magnets relative to the stationary coordinate system of the stator. Additionally, in the case of position or speed control, the output of the position sensor provides a feedback signal to the drive control system. The research carried out included the implementation of ten hardware modules for communication with a position sensor of a quadrature interface and Hall sensors with various levels of complexity. The modules allow for the reading of the following:

- Relative angle—rotor position given within one full revolution (from 0 to 2π), relative to the initial rotor position (at the moment of startup);
- Absolute angle—rotor position is given in the range of one revolution (from 0 to 2π); position 0 is set based on the third sensor channel (index);
- Relative position—rotor position, relative to the initial rotor position (at the moment of startup);
- Absolute position—rotor position; position 0 is set based on the third sensor channel (index);
- Angular velocity—the value of the current angular velocity, measured in two modes (as a position difference in a fixed period or as a time measured between two pulses), and then the value with the lowest error is selected; the module is equipped with a fourth-order low-pass filter;
- Electric angle—the angle of the position of the magnetic flux vector coming from permanent magnets when Hall sensors are used.

To implement the electronic commutation algorithm, a hardware block was designed, the task of which is to transform the mechanical angle (position of the shaft) to the electrical angle (position of the magnetic flux vector of permanent magnets) with a standardized value of 2^{16} (where 2^{16} corresponds to 2π). By normalizing the electrical angle to a constant value independent of the resolution of the position sensor used, the electronic commutation algorithm is simplified and unified. Another element of the control system is the regulator module, which consists of two regulators of proportional type. The set values for the controllers are the input value for i_{qref} and 0 for the i_d component. The values calculated in the controllers are then transformed into a polar coordinate system to adapt the values to the SVPWM modulator inputs. An additional element of the controller module is an interface for communication using a dedicated internal DDMC-BUS bus to set controller settings and read parameter values online. The lack of connection of the interface to the DDMC-BUS bus results in the removal of circuits used for communication in the optimization process during compilation. As a result, the hardware resources of the FPGA are not used when the interface is not in use.

The Clarke and Park transformation module is designed to transform the values of phase currents to values expressed in the rotating coordinate system (d, q) associated with the rotor. It consists of three internal components:

- An implemented CORDIC (COordinate Rotation DIgital Computer) method to determine the values of trigonometric sine and cosine functions;
- Implemented sequential multiplication methods allowing the obtaining the product of two arguments in the two's complement representation (U2);
- A state machine (Moore's finite machine), the task of which is to take input values (phase currents), determine the transform using the following components: CORDIC, sequential multiplication, and basic arithmetic operations, and return the final result.

To ensure the universality of the module's usage in various families of reprogrammable systems, only basic logical elements were used for its implementation without the application of built-in multipliers (DSP) and phase loops (PLL).

PWM modulation is implemented by SVPWM and PWM-HL blocks. SVPWM is responsible for determining the fill factors for each of the drive phases. The input parameters are the modulation coefficient and correction angle (the values are determined by the regulator module) and the position angle of the permanent magnet flux vector (coming from the position sensor). The output values are the fill factors for each drive phase. The module

uses the CORDIC algorithm to calculate the value of the sine function. PWM-HL generates signals intended to control the power stage, taking into account its structure (separate control of the upper and the lower half-bridge transistors) and modulation parameters (e.g., pulse position, modulation frequency, minimum and maximum duty cycles, so-called “dead times”) based on the fill factors determined in the SVPWM.

Table 1 shows the usage of hardware resources of the Cyclone IV—EP4CE22F17C6N FPGA system depending on the assumed computational accuracy and for the key hardware modules implementing the drive control algorithm. Additionally, the last column of the table shows the maximum clock frequency for each component. The module handling the incremental encoder is presented in two versions, with different levels of FPGA hardware resource usage. EncoInc(#1) is a version of the module in which only the output in the form of the absolute angle value is used (the knowledge of the absolute angle value is necessary to carry out the electronic commutation process of the drive). The second version of the module—EncoInc(#2)—is a version that allows for reading the relative angle, absolute angle, relative position, absolute position, and angular velocity. However, different levels of hardware resource usage do not mean that we deal with two various modules. Both modules have been implemented based on the same source code in the VHDL language. The difference in the usage of the logical elements and registers results from the optimization process carried out at the stage of project compilation in the Quartus Prime software (<https://www.intel.com/content/www/us/en/products/details/fpga/development-tools/quartus-prime/resource.html>), thanks to which the unused module structures (e.g., for determining angular velocity) were removed from the architecture. A similar process is visible in the case of the RegP regulator, in which the RegP(#2) version has an interface for communication via the internal DDMC-BUS bus (making it possible to write/read the regulator parameters online), while in the RegP(#1), version this interface has been removed at the compilation stage.

Table 1. Usage of the hardware resources of the FPGA system (Cyclone IV—EP4CE22F17C6N) for the main hardware modules.

Modules	Comput. Accuracy	Logic Elements (LEs)	Registers	Memory Bits	DSP Blocks	Max. Freq. [MHz]
EncoInc(#1)	6 bits	95 (0.4%)	31 (0.1%)	0 (0%)	0 (0%)	218.3
EncoInc(#2)	6 bits	844 (3.8%)	519 (2.3%)	0 (0%)	0 (0%)	161.4
EAngleNorm	6 bits	131 (0.6%)	120 (0.5%)	0 (0%)	0 (0%)	287.4
ClarkParkTrans	6 bits	484 (2.2%)	212 (0.9%)	0 (0%)	0 (0%)	149.1
RegP(#1)	6 bits	59 (0.3%)	2 (<0.1%)	0 (0%)	0 (0%)	1488.4
RegP(#2)	6 bits	320 (1.4%)	91 (0.4%)	0 (0%)	4 (3.0%)	715.3
SVPWM	6 bits	506 (2.3%)	180 (0.8%)	0 (0%)	0 (0%)	151.7
PWM-HL	6 bits	548 (2.5%)	203 (0.9%)	0 (0%)	0 (0%)	254.2
EncoInc(#1)	8 bits	95 (0.4%)	31 (0.1%)	0 (0%)	0 (0%)	218.3
EncoInc(#2)	8 bits	844 (3.8%)	519 (2.3%)	0 (0%)	0 (0%)	161.4
EAngleNorm	8 bits	132 (0.6%)	122 (0.5%)	0 (0%)	0 (0%)	296.7
ClarkParkTrans	8 bits	580 (2.6%)	262 (1.2%)	0 (0%)	0 (0%)	146.1
RegP(#1)	8 bits	119 (0.5%)	2 (<0.1%)	0 (0%)	0 (0%)	1420.5
RegP(#2)	8 bits	451 (2.0%)	99 (0.4%)	0 (0%)	4 (3.0%)	716.9
SVPWM	8 bits	593 (2.7%)	224 (1.0%)	0 (0%)	0 (0%)	144.8
PWM-HL	8 bits	665 (3.0%)	209 (0.9%)	0 (0%)	0 (0%)	262.5
EncoInc(#1)	10 bits	95 (0.4%)	31 (0.1%)	0 (0%)	0 (0%)	218.3
EncoInc(#2)	10 bits	844 (3.8%)	519 (2.3%)	0 (0%)	0 (0%)	161.4
EAngleNorm	10 bits	131 (0.6%)	124 (0.5%)	0 (0%)	0 (0%)	303.2
ClarkParkTrans	10 bits	706 (3.2%)	315 (1.4%)	0 (0%)	0 (0%)	140.7
RegP(#1)	10 bits	203 (0.9%)	2 (<0.1%)	0 (0%)	0 (0%)	1420.5
RegP(#2)	10 bits	640 (2.9%)	107 (0.5%)	0 (0%)	8 (6.1%)	715.3
SVPWM	10 bits	732 (3.3%)	271 (1.2%)	0 (0%)	0 (0%)	138.5
PWM-HL	10 bits	677 (3.0%)	209 (0.9%)	0 (0%)	0 (0%)	298.7
EncoInc(#1)	12 bits	95 (0.4%)	31 (0.1%)	0 (0%)	0 (0%)	218.3
EncoInc(#2)	12 bits	844 (3.8%)	519 (2.3%)	0 (0%)	0 (0%)	161.4
EAngleNorm	12 bits	133 (0.6%)	126 (0.6%)	0 (0%)	0 (0%)	281.5
ClarkParkTrans	12 bits	825 (3.7%)	368 (1.6%)	0 (0%)	0 (0%)	127.1

Table 1. Cont.

Modules	Comput. Accuracy	Logic Elements (LEs)	Registers	Memory Bits	DSP Blocks	Max. Freq. [MHz]
RegP(#1)	12 bits	311 (1.4%)	2 (<0.1%)	0 (0%)	0 (0%)	1420.5
RegP(#2)	12 bits	825 (3.7%)	115 (0.5%)	0 (0%)	8 (6.1%)	713.8
SVPWM	12 bits	852 (3.8%)	318 (1.4%)	0 (0%)	0 (0%)	125.9
PWM-HL	12 bits	594 (2.7%)	203 (0.9%)	0 (0%)	0 (0%)	279.0
EncoInc(#1)	14 bits	95 (0.4%)	31 (0.1%)	0 (0%)	0 (0%)	218.3
EncoInc(#2)	14 bits	844 (3.8%)	519 (2.3%)	0 (0%)	0 (0%)	161.4
EAngleNorm	14 bits	138 (0.6%)	130 (0.6%)	0 (0%)	0 (0%)	269.3
ClarkParkTrans	14 bits	918 (4.1%)	418 (1.9%)	0 (0%)	0 (0%)	123.4
RegP(#1)	14 bits	976 (4.4%)	3 (<0.1%)	0 (0%)	0 (0%)	1420.5
RegP(#2)	14 bits	1053 (4.7%)	123 (0.5%)	0 (0%)	8 (6.1%)	677.5
SVPWM	14 bits	944 (4.2%)	362 (1.6%)	0 (0%)	0 (0%)	124.0
PWM-HL	14 bits	638 (2.9%)	209 (0.9%)	0 (0%)	0 (0%)	269.5
EncoInc(#1)	16 bits	95 (0.4%)	31 (0.1%)	0 (0%)	0 (0%)	218.3
EncoInc(#2)	16 bits	844 (3.8%)	519 (2.3%)	0 (0%)	0 (0%)	161.4
EAngleNorm	16 bits	139 (0.6%)	134 (0.6%)	0 (0%)	0 (0%)	295.4
ClarkParkTrans	16 bits	1073 (4.8%)	474 (2.1%)	0 (0%)	0 (0%)	116.4
RegP(#1)	16 bits	1263 (5.7%)	3 (<0.1%)	0 (0%)	0 (0%)	1420.5
RegP(#2)	16 bits	1290 (5.8%)	131 (0.6%)	0 (0%)	8 (6.1%)	684.0
SVPWM	16 bits	1102 (4.9%)	408 (1.8%)	0 (0%)	0 (0%)	113.7
PWM-HL	16 bits	643 (2.9%)	209 (0.9%)	0 (0%)	0 (0%)	288.5

Based on the data gathered in Table 1, it can be stated that the hardware modules that use the hardware resources are the Clarke and Park transforms, the regulator, and the SVPWM modulation. Such a situation results from the implementation of numerical calculations in these modules. The changes in the assumed computational accuracy significantly affect the hardware resources used in the reprogrammable system. In the case of modules for operating the incremental encoder and normalizing the electrical angle, the assumed computational accuracy practically does not affect the amount of resources used in the system. It can also be noticed that the increase in the number of bits on which the operations are performed results in a reduction of the maximum operating frequency of the module.

3.3. DDSApp Application

DDSApp application consists of a part of a system for investigating direct drive commutation algorithms. It has two interfaces intended for communication with the test stand controller and the controller of the tested drive implemented in the FPGA system. The application allows for saving and reading operating parameters and measurement data. The use of the USB bus in the virtual serial port mode makes it possible to achieve a maximum transmission speed of up to 6 MBd and, as a result, to transmit the measurement data in each sampling period of the drive controller. Additionally, the application supports saving all measurement data as a text file, which allows for the subsequent analysis and presentation of the results in the MATLAB environment for numerical computations.

Communication with the test stand controller is mainly used to set the operating parameters of the functional waveform generator and to acquire measurement data during tests (including the signal from the torque sensor, set trajectory, load torque, and braking torque). Communication with the controller of the drive under test allows for writing and reading the controller operating parameters (e.g., regulator settings) and acquiring measurement data. The program is designed to handle eleven types of data frames consisting of a preamble, sample number (frame number), measurement data (from one to eight channels) in the format of fixed-point (32 bit) or single-precision floating-point values (float), and the checksum.

The graphical user interface of the DDSApp application consists of a main window intended for data presentation (charts, parameter values, instantaneous values of waveforms, maximum and average errors between selected pairs of channels) and a side panel intended for setting parameters. The side panel makes it possible to achieve the following:

- Set the data transmission parameters between the application and the test stand controller (transmission speed, data frame type);
- Set the data transmission parameters between the application and the drive controller (transmission speed, data frame type);
- Set the measurement data display parameters (sampling period, window width, displayed channels);
- Set the parameters of the controller's function waveform generator;
- Set the operating parameters of the drive controller (e.g., settings of regulators);
- Set the parameters of saving data to the local disk (e.g., file name).

4. Experimental Research

Experimental research aimed to determine the impact of the computational accuracy used in the electronic commutation algorithms of direct drives on the ripples, the rate of torque changes, as well as position-holding and trajectory-tracking accuracies. The term "computational accuracy" means performing arithmetic operations on the fixed-point numbers in the field-oriented control method on vectors with a specific bit length: from 6 bit to 16 bit. In the electronic commutation system, the SVPWM modulation was used, while the operating frequency of the current loop and PWM modulation amounted to 12.2 kHz. The research was carried out by implementing an electronic commutation system for the direct drive and comparing the impact of changes in the computational accuracy while keeping the remaining control parameters unchanged (control frequency, PWM modulation frequency, PWM modulation type).

A proportional controller (P-type) is used in the current and position loops of the drive control system. The selection of such a controller type resulted from the necessity to maintain a uniform procedure for selecting controller settings for systems of different dynamic properties. The more complex controllers (e.g., PID) would make it difficult to maintain a uniform criterion for selecting the controller settings.

To assess the accuracy of time history mapping and the error values, three measures were selected: maximum error, average error, and root mean square error. The maximum error determines the maximum value of the difference between the set value and the measured value for the analyzed time history and is defined as follows:

$$\text{MaxErr} = \max_{i=1}^N (|s_i - c_i|), \quad (4)$$

where s denotes the time history of the set value of the length equal to N samples, and c is the time history of the measured value of the length equal to N samples. The mean error was defined as the average of the absolute values of the error function:

$$\text{AvgErr} = \frac{1}{N} \sum_{i=1}^N |s_i - c_i|. \quad (5)$$

The root mean square error was defined as

$$\text{RMSErr} = \sqrt{\frac{1}{N} \sum_{i=1}^N (s_i - c_i)^2} \quad (6)$$

It was chosen due to the greater sensitivity of this coefficient to high-amplitude oscillations and differences between the set and measured values compared to the average error.

4.1. Static Experiment

The static experiment consisted of implementing a direct drive control system of the PMSM type in the torque (current) mode and examining the system response to the set torque in the form of a sequence of bipolar rectangular pulses with the electric brake clamped (immobilized) (test at a rotational speed of 0 rpm). The setting of the P-type (proportional) controller of the I_q component of the stator current has been selected to obtain the shortest rise time of the I_q component slope of the stator current, preserving

the inertial character of the system response (without overregulation). The setting of the P-type controller of the I_d component of the stator current was selected to obtain the lowest average error of this component. Figure 7 shows an example of the set current characteristic as the sequence of the bipolar rectangular impulses and the system response in the form of the I_d and I_q components of the stator current.

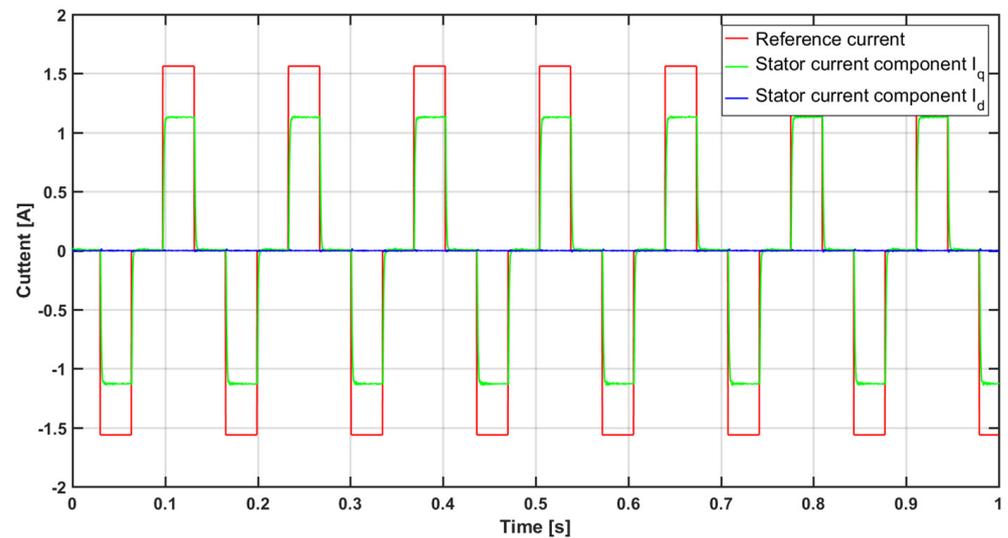


Figure 7. Time histories of rectangular impulses.

The results of all static experiments presented below are the average from a series of 12 measurements. Each measurement series was performed for the same rotor positions (angles) spaced by 30° within one electrical revolution. Such a proceeding made it possible to reduce the impact of the asymmetric distribution of the drive magnetic flux resulting from local disturbances in the distribution of the rotor magnetic field and inaccuracies in the winding of the stator on the obtained measurement results.

In the course of the static experiments carried out, for the purposes of assessing the impact of the operating parameters of the direct drive control system on the ripples and the rate of changes of the torque and ripples of the I_q and I_d components of the stator current, the following indicators (measures) have been determined:

- t_{edge} —the average rise and fall time of the I_q component of the stator current, expressed as the rise or fall time of the edge between 10% and 90% of the steady-state value of the I_q component of the stator current (Figure 8);
- t_{del} —delay time, expressed as the time between the rising (or falling) of the edge of the set current and the point of reaching 10% of the set value of the I_q component of the stator current (Figure 8);
- I_{qsv} —the steady-state value of the I_q component of the stator current determined as the arithmetic mean of the second half of the duration of the high (or low) state of the rectangular impulse (Figure 8);
- SR_q —the average rate of rising and falling of the edges of the I_q component of the stator current, expressed as the quotient of 80% of the steady-state value of the I_q current component and the t_{edge} time;
- $I_{qsv}^{\%}$ —the steady-state value of the I_q component of the stator current concerning the set value;
- I_q^{MaxErr} —maximal error of the I_q component of the stator current determined in the second half of the duration of the high state of the rectangular impulse, determining the maximal torque ripple (Figure 8);
- I_q^{AvgErr} —average value of the error of the I_q component of the stator current determined in the second half of the duration of the high (low) state of the rectangular impulse, determining the average torque ripples (Figure 8);

- I_q^{RMSErr} —the root mean square error of the I_q component of the stator current determined in the second half of the duration of the high (or low) state of the rectangular impulse, determining the root mean square value of the torque ripples;
- I_d^{MaxErr} —the maximal value of error of the I_d component of the stator current, determined over the entire measurement period (Figure 8);
- I_d^{AvgErr} —average value of error of the I_d component of the stator current, determined over the entire measurement period;
- I_d^{RMSErr} —the root mean square error of the I_d component of the stator current, determined over the entire measurement period;
- T^{Max} —the maximum value of torque ripples determined in the second half of the duration of the high (or low) state of the square impulse, based on data from the torque sensor;
- T^{Avg} —the average value of torque ripples determined in the second half of the duration of the high (or low) state of the square impulse, based on data from the torque sensor;
- T^{RMS} —the root mean square error of the torque ripple determined in the second half of the duration of the high (or low) state of the square impulse, based on data from the torque sensor.

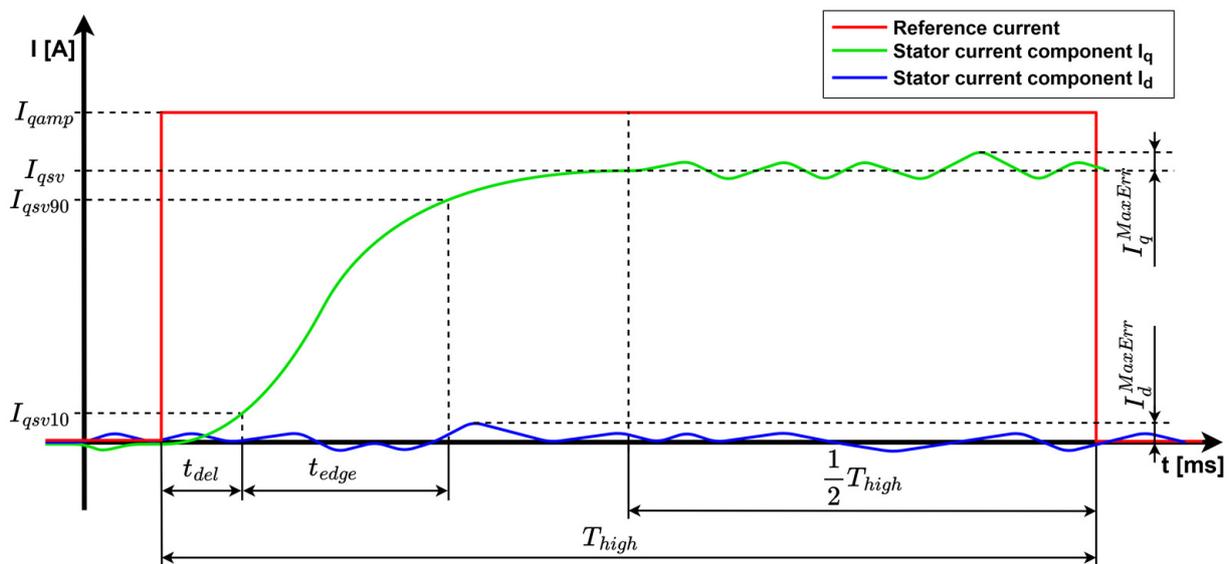


Figure 8. An example time history of the set (reference) current and stator current components for a static experiment. T_{high} —duration of the high state of the rectangular impulse; I_{qamp} —amplitude of the rectangular impulse; I_{qsv} —determined value of the I_q component of the stator current; I_{qsv10} —10% of the I_{qsv} value; I_{qsv90} —90% of the I_{qsv} value; I_q^{MaxErr} —maximal error of the I_q component; I_d^{MaxErr} —maximal error of the I_d component.

Based on the output signal of the torque sensor, the maximum, average, and root mean square values of the torque ripples were determined. Evaluation of the times of rising and falling of the slopes and of the rate of the observed changes was not possible due to the oscillations resulting from the application of the claw clutches with flexible inserts between the tested drive and the torque sensor as well as between the torque sensor and the electric brake. An example time history of the torque (obtained from the torque sensor) is presented in Figure 9. Despite using connectors with the highest available hardness, a sudden change in torque resulted in oscillations. The primary benefit of implementing direct drives is the ability to circumvent the use of clutches and gears in mechanical design, thereby eliminating issues related to clearances and stiffness maintenance.

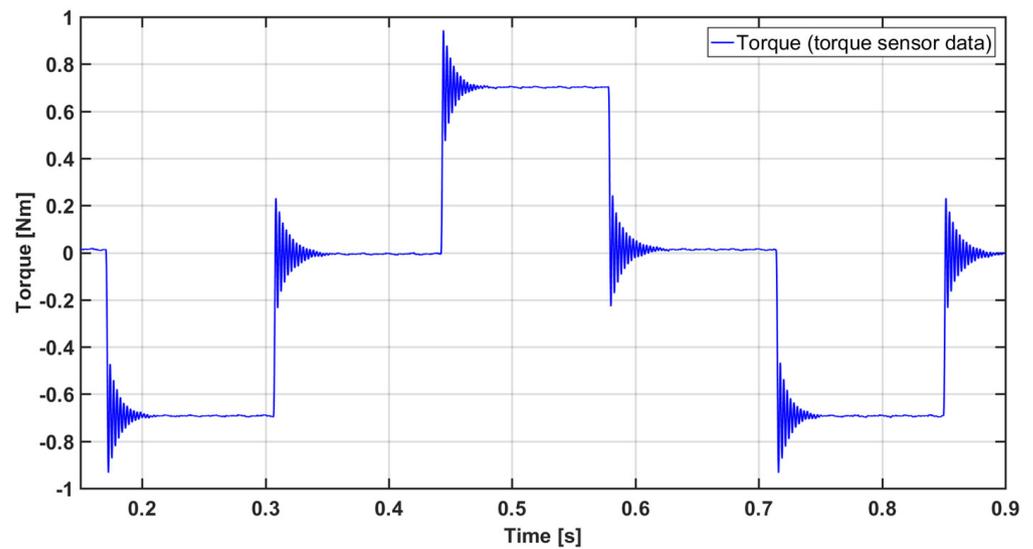


Figure 9. Time history of the signal from the torque sensor.

4.2. Dynamic Experiment

The second type of experimental research that was carried out consisted of dynamic experiments aiming to determine the accuracy of position holding and tracking the trajectory. The results of dynamic experiments made it possible to establish the influence of the parameters of algorithms used in the electronic commutation system of direct drives on the quality of control. The external torque generated by the second drive of the test stand based on the given characteristic was used as the load for the tested drive. Additional inertial loads resulted from the presence of the rotors of both drives, the clutch, and the auxiliary shaft, with the total moment of inertia equal to $15 \text{ kg}\cdot\text{cm}^2$. The operating frequency of the position loop in each case was equal to half of the operating frequency of the electronic commutation system, i.e., 6.1 kHz. The setting of the P-type regulator of the position loop of the drive control system was selected to achieve the lowest value of the maximum error for the entire trajectory. To determine the positioning and tracking errors for the trajectory in a wide range of drive operations, three characteristics with different dynamics were prepared: slow, fast, and very fast.

The trajectory of the slow waveform consisted of five repetitions of a trapezoidal signal, a sinusoidal signal with saturation, and a sinusoidal signal, each with an amplitude of 540° and a period of 5 s. The last section of the trajectory had constant values and lasted 20 s, which allowed for checking the errors in holding the given position. Using the trajectory consisting of fragments (segments) of different characters made it possible to check the behavior of the controller both for mild waveforms with all derivatives continuing and C_0 class waveforms (having a discontinuous derivative). The external load generated by the second drive was a sinusoidal torque with an amplitude of 1 Nm (which amounted to 79% of the rated torque of the tested drive) and a frequency of 2 Hz.

The fast waveform had a trajectory similar in shape to the slow waveform, with an amplitude of 180° and a period of 1 s. As an external load, the sinusoidal moment with an amplitude of 0.5 Nm and a frequency of 10 Hz was used. The set trajectory and load for the fast waveform are presented in Figure 10.

The very fast waveform had a trajectory with an amplitude of 36° and a period of 0.2 s. As an external load, the sinusoidal moment with an amplitude of 0.5 Nm and a frequency of 50 Hz was used.

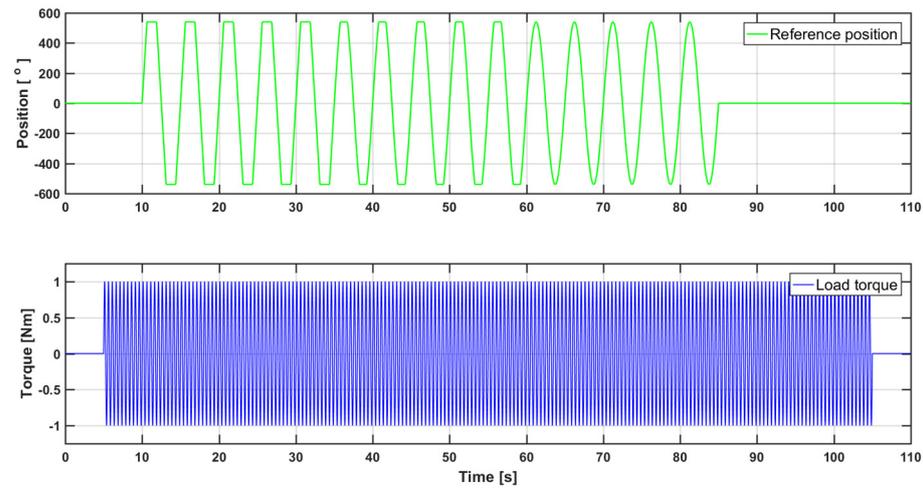


Figure 10. Set trajectory and set load torque for the slow characteristic.

5. Results

5.1. Static Experiment

Table 2 shows values of the direct drive control quality measures (indicators) determined based on a static experiment concerning the computational accuracy used (from 6 to 16 bits). The following parameters were compared by applying the methodology presented in papers [40,41]: the average times of rising (and falling) of the edges of the I_q component of the stator current, the average delay times of the I_q component of the stator current, the ripples of both components of the stator current, the steady-state values of the I_q component of the stator current, and the rates of rising (and falling) of the edges of the I_q component of the stator current.

Table 2. Values of the average times of rising (and falling) of the edges, delay times, ripples of the I_q and I_d components of the stator current, the steady-state values of the current, and the rates of rising (and falling) of the edges of the stator current components with respect to the computational accuracy used.

Comput. Accuracy	t_{edge} [ms]	t_{del} [ms]	Errors of I_q Component [mA]			Errors of I_d Component [mA]			$I_{qsv}^{\%}$ [%]	SR_q [A/ms]
			I_q^{MaxErr}	I_q^{AvgErr}	I_q^{RMSErr}	I_q^{MaxErr}	I_q^{AvgErr}	I_q^{RMSErr}		
6 bits	1.40	0.38	261.13	68.66	90.22	325.52	73.64	121.28	75.89	0.69
8 bits	1.42	0.41	52.46	15.97	21.33	85.45	8.49	20.18	75.62	0.67
10 bits	1.39	0.41	19.41	5.05	6.60	34.59	4.00	7.39	75.77	0.68
12 bits	1.38	0.41	14.48	2.78	3.55	30.26	5.08	6.66	75.67	0.69
14 bits	1.37	0.41	13.33	2.68	3.40	24.60	3.53	4.56	75.50	0.69
16 bits	1.39	0.41	13.03	2.52	3.20	22.09	3.73	4.64	75.80	0.68

Table 3 shows values of the rotational torque ripples and the set rotational torque value for direct drive control implemented with different computational accuracy.

Table 3. Values of the rotational torque ripples, the steady-state value of the rotational torque, and the torque constant of the drive with respect to the computational accuracy used.

Comput. Accuracy	Ripples of the Rotational Torque [mNm]			T_{SV} [mNm]
	$T^{MaxRipp}$	$T^{AvgRipp}$	$T^{RMSRipp}$	
6 bits	46.14	8.52	11.49	372.16
8 bits	18.91	2.96	3.96	343.69
10 bits	14.39	1.87	2.43	350.31
12 bits	14.06	1.75	2.30	348.90
14 bits	14.29	1.78	2.34	345.23
16 bits	14.16	1.75	2.31	345.93

Figures 11–13 contain values from Tables 1 and 2 presenting the ripples of the stator current components and the rotational torque ripples (based on the data from the torque sensor), depending on the computational accuracy used. In all presented cases, the error values decree with computational accuracy.

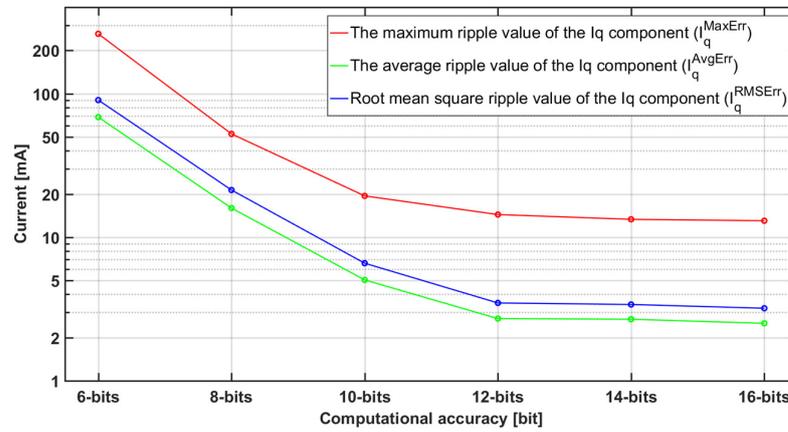


Figure 11. Ripples of the I_q component of the stator current with respect to the computational accuracy used.

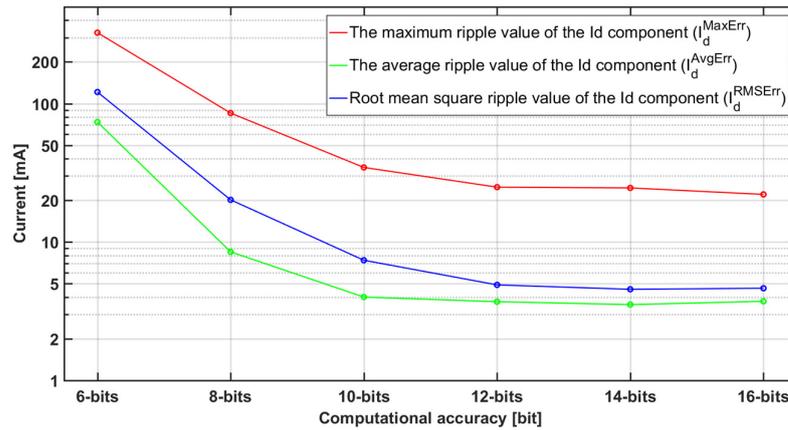


Figure 12. Ripples of the I_d component of the stator current with respect to the computational accuracy used.

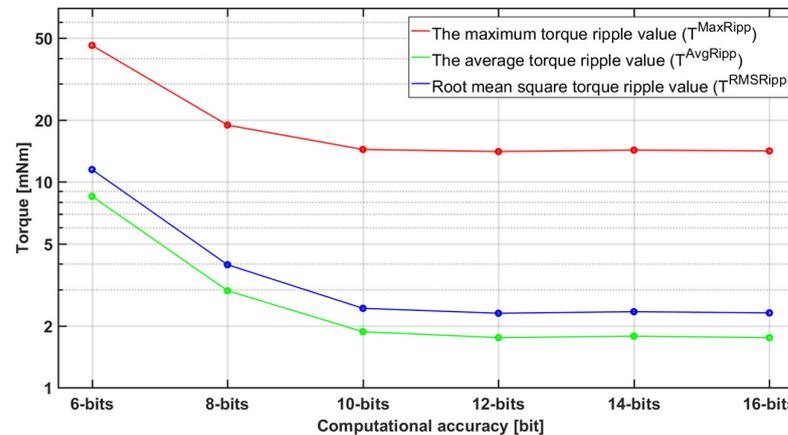


Figure 13. Ripples of the rotational torque with respect to the computational accuracy used.

Table 4 shows the usage of the hardware resources of the FPGA system necessary to implement the electronic commutation system, depending on the computational accuracy used for the static experiment.

Table 4. Usage of the hardware resources of the FPGA system (Cyclone IV—EP4CE22F17C6N) for the implementation of the electronic commutation system with respect to the computational accuracy used.

Comput. Accuracy	Logic Elements (LEs)	Registers	Memory Bits	DSP Blocks
6 bits	1768 (8%)	742 (3%)	0 (0%)	0 (0%)
8 bits	2072 (9%)	833 (4%)	0 (0%)	0 (0%)
10 bits	2411 (11%)	927 (4%)	0 (0%)	0 (0%)
12 bits	2686 (12%)	1018 (5%)	0 (0%)	0 (0%)
14 bits	3385 (15%)	1022 (5%)	0 (0%)	0 (0%)
16 bits	3977 (18%)	1124 (5%)	0 (0%)	0 (0%)

5.2. Dynamic Experiment

Tables 5–10 present errors in position holding and trajectory tracking for slow, fast, and very fast waveforms, along with the specification of trajectory-tracking errors for trapezoidal, saturated sinusoidal, and sinusoidal sections (segments).

Table 5. Trajectory-tracking and position-holding errors for a slow waveform depending on the computational accuracy used.

Comput. Accuracy	Position-Holding Errors [°]			Trajectory-Tracking Errors [°]		
	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}
6 bits	3.1	1.49	1.65	8.5	1.62	1.94
8 bits	2.92	1.59	1.74	8.68	1.74	2.09
10 bits	3.1	1.57	1.72	8.42	1.71	2.05
12 bits	3.2	1.58	1.72	8.96	1.74	2.09
14 bits	3.06	1.57	1.72	8.6	1.73	2.07
16 bits	2.52	1.57	1.71	8.42	1.69	2

Table 6. Trajectory-tracking errors for the consecutive segments of a slow waveform depending on the computational accuracy used.

Comput. Accuracy	Trapezoidal Signal [°]			Sinusoidal Signal with Saturation [°]			Sinusoidal Signal [°]		
	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}
6 bits	8.42	1.81	2.26	4.57	1.56	1.82	4.75	1.55	1.81
8 bits	8.6	1.98	2.49	5.04	1.7	1.99	4.64	1.63	1.89
10 bits	8.42	1.95	2.45	4.9	1.65	1.93	4.68	1.6	1.85
12 bits	8.96	2.01	2.53	4.97	1.66	1.95	4.68	1.62	1.88
14 bits	8.6	1.97	2.47	5.08	1.67	1.95	4.61	1.63	1.88
16 bits	8.42	1.87	2.32	4.97	1.64	1.92	4.21	1.61	1.84

Table 7. Trajectory-tracking and position-holding errors for a fast waveform depending on the computational accuracy used.

Comput. Accuracy	Position-Holding Errors [°]			Trajectory-Tracking Errors [°]		
	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}
6 bits	29.38	17.06	18.97	58.32	16.79	19.71
8 bits	27.11	16.21	18	55.87	16.04	18.83
10 bits	27.61	16.49	18.31	55.84	16.29	19.13
12 bits	27.25	16.42	18.22	55.76	16.23	19.05
14 bits	27.47	16.58	18.4	54.54	16.35	19.12
16 bits	27.68	16.95	18.81	55.51	16.51	19.25

Table 8. Trajectory-tracking errors for the consecutive segments of a fast waveform depending on the computational accuracy used.

Comput. Accuracy	Trapezoidal Signal [°]			Sinusoidal Signal with Saturation [°]			Sinusoidal Signal [°]		
	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}
6 bits	58.32	18	22.31	39.06	17.07	19.58	32.26	16.9	18.99
8 bits	53.32	17.17	21.05	38.48	16.32	18.84	31.28	16.08	18.12
10 bits	54.32	17.46	21.44	38.38	16.62	19.15	31.61	16.35	18.4
12 bits	53.89	17.41	21.34	38.34	16.54	19.06	31.79	16.28	18.32
14 bits	54.11	17.56	21.48	38.88	16.73	19.26	32.22	16.46	18.52
16 bits	55.15	17.68	21.52	38.95	16.98	19.48	32.62	16.72	18.81

Table 9. Trajectory-tracking and position-holding errors for a very fast waveform depending on the computational accuracy used.

Comput. Accuracy	Position-Holding Errors [°]			Trajectory-Tracking Errors [°]		
	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}
6 bits	0.65	0.19	0.24	22.32	2.5	4.59
8 bits	0.4	0.15	0.17	22.72	2.11	4.13
10 bits	0.36	0.15	0.17	22.9	2.46	4.38
12 bits	0.36	0.15	0.17	22.9	2.42	4.41
14 bits	0.32	0.15	0.17	22.79	2.4	4.5
16 bits	0.4	0.15	0.17	22.32	2.53	4.56

Table 10. Trajectory-tracking errors for the consecutive segments of a very fast waveform depending on the computational accuracy used.

Comput. Accuracy	Trapezoidal Signal [°]			Sinusoidal Signal with Saturation [°]			Sinusoidal Signal [°]		
	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}	ϕ^{MaxErr}	ϕ^{AvgErr}	ϕ^{RMSErr}
6 bits	18.83	6.48	8.23	16.34	5.51	6.83	4	1.48	1.8
8 bits	13.93	2.97	4.5	18.4	4.99	6.62	3.42	1.43	1.7
10 bits	15.73	5.62	6.94	15.08	5.26	6.51	6.08	1.73	2.15
12 bits	15.7	5.6	6.97	18.61	5.42	7.06	5.69	1.68	2.07
14 bits	14.8	4.84	6.65	20.23	5.74	7.59	5.94	1.74	2.15
16 bits	18.04	5.94	7.52	18.68	5.34	6.92	6.8	1.84	2.31

Figures 14–19 present the errors (maximum, average, and root mean square error) of trajectory tracking and position holding, depending on the computational accuracy used, for slow, fast, and very fast waveforms.

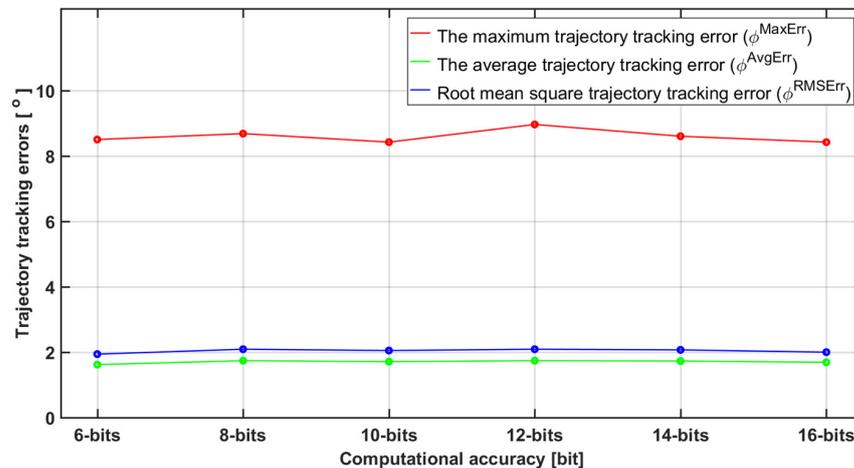


Figure 14. Errors of trajectory tracking for a slow waveform with respect to the computational accuracy used.

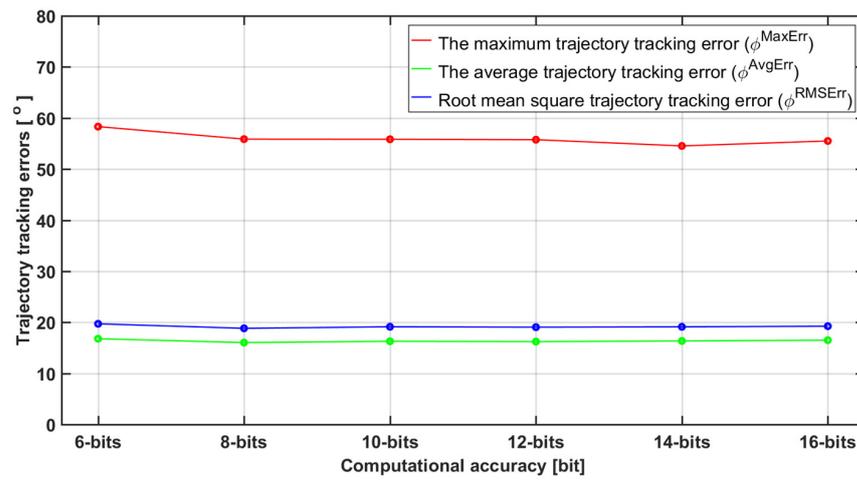


Figure 15. Errors of trajectory tracking for a fast waveform with respect to the computational accuracy used.

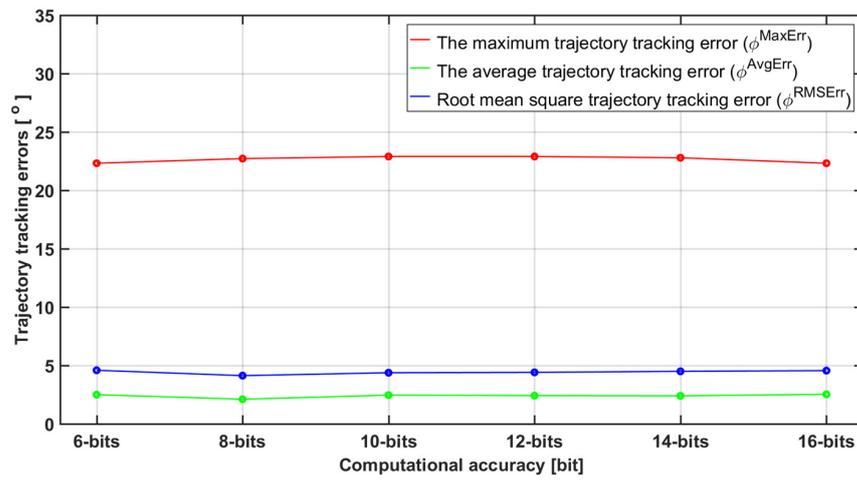


Figure 16. Errors of trajectory tracking for a very fast waveform with respect to the computational accuracy used.

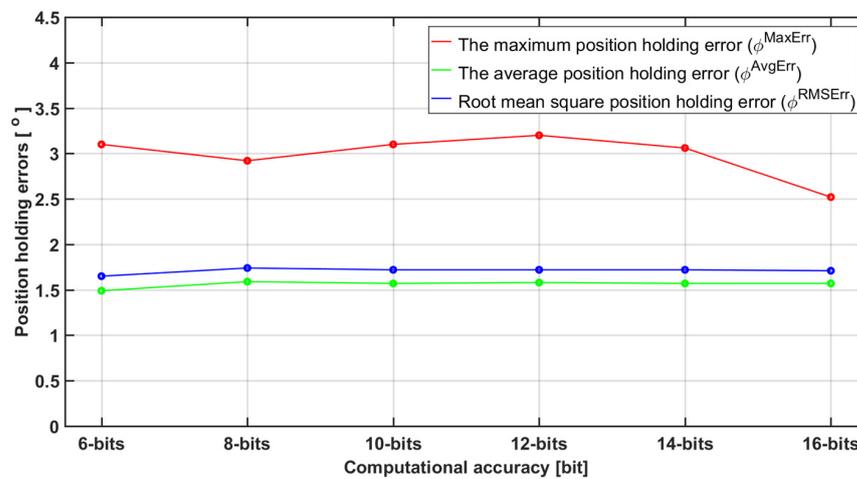


Figure 17. Position-holding errors for slowly changing load, determined with respect to the computational accuracy used.

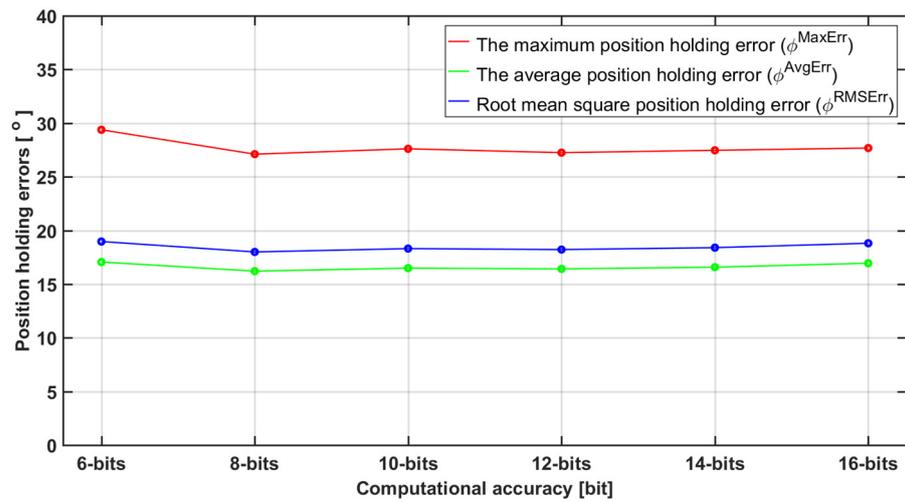


Figure 18. Position-holding errors for fast-changing load, determined with respect to the computational accuracy used.

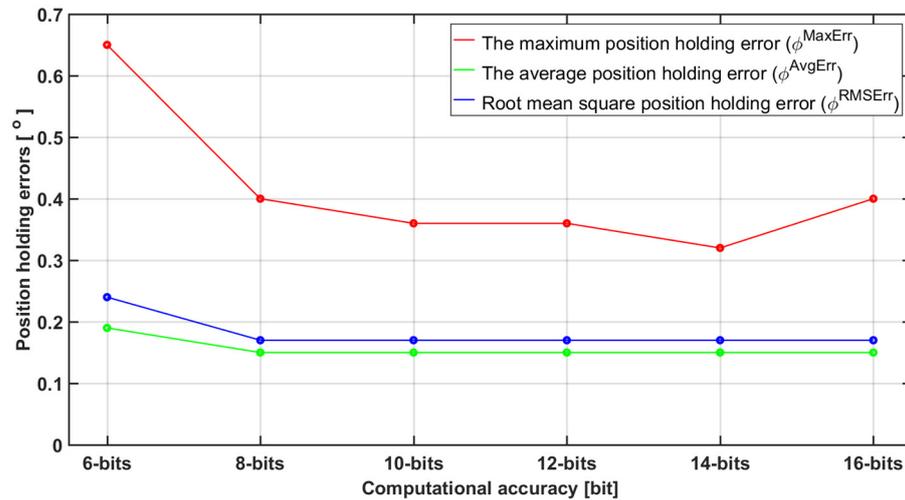


Figure 19. Position-holding errors for very fast-changing load, determined with respect to the computational accuracy used.

Table 11 shows the usage of hardware resources by the FPGA system necessary to implement the position controller, depending on the computational accuracy used in the dynamic experiment.

Table 11. Usage of the hardware resources of the FPGA system (DE0-Nano EP4CE22F17C6N) required to implement the position controller, depending on the computational accuracy used.

Comput. Accuracy	Logic Elements (LEs)	Registers	Memory Bits	DSP Blocks
6 bits	1874 (8%)	775 (3%)	0 (0%)	4 (3%)
8 bits	2181 (10%)	866 (4%)	0 (0%)	4 (3%)
10 bits	2522 (11%)	960 (4%)	0 (0%)	4 (3%)
12 bits	2815 (13%)	1051 (5%)	0 (0%)	4 (3%)
14 bits	3495 (16%)	1055 (5%)	0 (0%)	4 (3%)
16 bits	4119 (18%)	1157 (5%)	0 (0%)	4 (3%)

6. Discussion

The presented results of the static experiment indicate that for the analyzed direct drive control system, the ripples of both the stator current components and the torque

decrease exponentially with the increase in computational accuracy. For the accuracy above 12 bits, the ripples of the stator current components and rotational torque change to a small extent. The average times of the rise (and fall) of the edges of the I_q component of the stator current, the average delay times, and the rate of rise (and fall) of the edges of the I_q component of the stator current does not depend on the computational accuracy used in the electronic commutation system. The steady-state value of the I_q component of the stator current, expressed concerning the set value, is also constant for all tested calculation accuracies.

The control system implemented with a 6-bit accuracy allows for the correct control of the direct drive, achieving maximum errors of the I_q component of the stator current of approximately 15% of the set value and the ripple of the I_d component at a similar level. The control system implemented with a 12-bit accuracy allows for a ten-fold reduction in the maximum and average ripples of the stator current components while increasing the usage of hardware resources of the FPGA system by just over 50% compared to the implementation with a 6-bit accuracy. The control system implemented with a 16-bit accuracy allows for reducing the ripples of the I_q component by approximately 10% and the ripples of the I_d component by approximately 20% compared to the implementation with a 12-bit accuracy while increasing the usage of hardware resources of the FPGA system by nearly 50%.

The results of the dynamic experiment include angle error values for holding position and trajectory tracking. Upon analyzing these values, no distinct trend emerges, whether indicating an increase or decrease as a function of computational accuracy. Considering the errors (maximum ϕ^{MaxErr} , average ϕ^{AvgErr} , and RMS ϕ^{RMSErr}) of maintaining position in Tables 5, 7 and 9, the difference between the extremes follows 0.68° , 0.1° , and 0.09° for slow; 2.27° , 0.85° , and 0.97° for fast; and 0.33° , 0.04° , and 0.07° for very fast changing loads. Similarly, the values arranged for trajectory-tracking errors are 0.54° , 0.12° , and 0.15° for slow; 3.34° , 0.75° , and 0.78° for fast; and 0.58° , 0.42° , and 0.46° for very fast waveforms. The listed values are not large, indicating that none of the developed controllers deviates significantly from the others. It is worth noting that these differences are not always calculated between the controllers with the 6-bit and 16-bit accuracy.

Another important factor related to trajectory tracking is the shape of the trajectory. As the results suggest, it has a determining effect on the level of errors. Referring to values in Tables 6, 8 and 10, the controller achieved the highest level of errors while tracking a trapezoidal trajectory. The smallest were obtained for a sinusoidal shape. It is regardless of the type (dynamics) of the waveform. Considering the waveform dynamics, it can be inferred that the minimal trajectory-tracking errors were observed for the slow waveform, whereas the maximum errors were observed for the fast waveform.

Based on the obtained results of the dynamic experiment, where the errors (maximum, average, and root mean square error) of position holding and trajectory tracking were analyzed, it can be concluded that for the analyzed direct drive control system (position controller with a P-type controller), the values of trajectory-tracking and position-holding errors are independent of the applied computational accuracy of the electronic commutation system. It means that adding an external feedback loop in the form of a position loop allows for obtaining similar results in position holding and trajectory tracking, regardless of the computational accuracy with which calculations are carried out in the current loop of the control system.

7. Conclusions

Based on the presented results, it can be concluded that satisfactory operating parameters for direct drive, including electronic commutation, can be obtained with relatively low accuracy in the numerical calculations of the commutation algorithm. The electronic commutation system implemented with a 6-bit accuracy allows for correct direct drive control. In the case of the direct drive working in the current (torque) mode, the ripples of the stator current components and the ripples of the rotational torque components decrease

exponentially, with computational accuracy increasing in the range from 6 to 12 bits. No notable enhancement in the operating parameters of the direct drive was observed for computations conducted on fixed-point numbers larger than 12 bits.

Implementation of the control algorithm in the current (torque) mode with a computational accuracy of 12 bits allows for a more than ten-fold reduction in the maximum and average ripples of the stator current components while increasing the usage of hardware resources of the FPGA system by over 50% compared to the implementation with a 6-bit accuracy.

In the experiments comparing the impact of computational accuracy on trajectory tracking and position holding, similar error values are obtained, regardless of the numerical calculation accuracy used, ranging from 6 bits to 16 bits. The introduction of an external feedback loop in the form of a position loop allows the implementation of the current loop algorithm with 6-bit accuracy without losing the quality of control parameters. As a result, it enables the implementation of the direct drive control system on hardware platforms with a lower level of complexity or using fewer hardware resources in the case of reprogrammable systems.

The reviewed literature lacks the results that would allow for comparing the impact of the computational accuracy of the electronic commutation system on the dynamics and ripples of the rotational torque stator current components, as well as trajectory-tracking and position-holding errors. The conducted research made it possible to indirectly determine the impact of the amount of hardware resources of the FPGA system used for implementation on the quality of direct drive control. Considering positional control, increasing computational accuracy does not significantly reduce trajectory-tracking and position-holding errors. However, it results in a significant increase in the usage of the FPGA hardware resources.

Author Contributions: Methodology, G.G., M.P., K.G., J.G., J.I. and M.M.; software, G.G.; validation, G.G., M.P. and K.G.; formal analysis, G.G., M.P., K.G., J.G., J.I., M.M. and W.Z.; investigation, G.G., M.P., K.G., J.G., J.I. and M.M.; resources, G.G., M.P., K.G., J.G., J.I., M.M. and W.Z.; data curation, G.G., M.P., K.G., J.G., J.I., M.M. and W.Z.; writing—original draft preparation, G.G., M.P., K.G., J.G., J.I., M.M. and W.Z.; writing—review and editing, G.G., M.P., K.G., J.G., J.I., M.M. and W.Z.; visualization, G.G., M.P. and K.G.; supervision, J.I.; project administration, J.I. and W.Z.; funding acquisition, J.I. and W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The research presented in this paper was partially financed by the INNOGLOBO/InnoIndie project “System for monitoring the conditions of transport of sensitive materials, including food and hazardous materials” (INNOGLOBO/I/132/InnoIndie/2022).

Data Availability Statement: Data will be made available upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

FPGA	Field-Programmable Gate Arrays
PMSM	Permanent Magnet Synchronous Motor
BLDC	Brushless Direct Current Motor
BEMF	Back Electromotive Force
FOC	Field-Oriented Control
DTC	Direct Torque Control
PWM	Pulse Width Modulation
DSP	Digital Signal Processors
DC	Direct Current
PI	Proportional-Integral

SPM	Surface Permanent Magnet
IPM	Interior Permanent Magnet
MTPA	Maximum Torque Per Ampere
MPC	Model Predictive Control
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
IGBT	Insulated Gate Bipolar Transistor
HDL	Hardware Description Language
RLC	RLC measurement bridge
MT	Tested Motor
TS	Torque Sensor
EB	Electric Brake
ML	Motor Load
MLD	Motor Load Drive Controller
SC	Test Stand Controller
MTI	Tested Motor Power Stage
MTC	Tested Motor Commutator
CP	Control Panel
PS	Power Supply
USB	Universal Serial Bus
TTL	Transistor-Transistor Logic
PCB	Printed Circuit Board
VHDL	Very High Speed Integrated Circuits Hardware Description Language
SVPWM	Space Vector Pulse Width Modulation
UART	Universal Asynchronous Receiver-Transmitter
SPI	Serial Peripheral Interface
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
CORDIC	Coordinate Rotation Digital Computer
PLL	Phase-Locked Loop
PID	Proportional-Integral-Derivative
P	Proportional
RMS	Root Mean Square

References

1. Cai, S.; Kirtley, J.L.; Lee, C.H.T. Critical Review of Direct-Drive Electrical Machine Systems for Electric and Hybrid Electric Vehicles. *IEEE Trans. Energy Convers.* **2022**, *37*, 2657–2668. [[CrossRef](#)]
2. Mohanraj, D.; Arulavid, R.; Verma, R.; Sathiyasekar, K.; Barnawi, A.B.; Chokkalingam, B.; Mihet-Popa, L. A Review of BLDC Motor: State of Art, Advanced Control Techniques, and Applications. *IEEE Access* **2022**, *10*, 54833–54869. [[CrossRef](#)]
3. Bogdan, V.; Adrian, M.; Leonard, L.; Alexandra, B.; Alecsandru, S.; Ionut, N. Design and Optimization of a BLDC Motor for Small Power Vehicles. In Proceedings of the 2021 International Conference on Electromechanical and Energy Systems (SIELMEN), Iasi, Romania, 6 October 2021; pp. 438–443. [[CrossRef](#)]
4. Gao, C.; Gao, M.; Si, J.; Hu, Y.; Gan, C. A Novel Direct-Drive Permanent Magnet Synchronous Motor with Toroidal Windings. *Energies* **2019**, *12*, 432. [[CrossRef](#)]
5. Sakunthala, S.; Kiranmayi, R.; Mandadi, P.N. A Study on Industrial Motor Drives: Comparison and Applications of PMSM and BLDC Motor Drives. In Proceedings of the 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017, Chennai, India, 19 June 2018; pp. 537–540. [[CrossRef](#)]
6. Bello, A.; Muhammad Kilishi, I.; Musa Bari, M.; Usman, A.; Abubakar, U. Comparative Review of PMSM And BLDCM Based on Direct Torque Control Method. *Int. J. Sci. Technol. Res.* **2014**, *3*, 195–199.
7. Surakasi, B.; Satish, R.; Pydi, B.; Kotb, H.; Shouran, M.; Abdul Samad, B. A Novel Methodology to Enhance the Smooth Running of the PM BLDC Motor Drive Using PWM-PWM Logic and Advance Angle Method. *Machines* **2023**, *11*, 41. [[CrossRef](#)]
8. Kolano, K.; Moradewicz, A.J.; Drzymala, B.; Geça, J. Influence of the Placement Accuracy of the Brushless DC Motor Hall Sensor on Inverter Transistor Losses. *Energies* **2022**, *15*, 1822. [[CrossRef](#)]
9. Shrutika, C.; Matani, S.; Chaudhuri, S.; Gupta, A.; Gupta, S.; Singh, N. Back-EMF Estimation Based Sensorless Control of Brushless DC Motor. In Proceedings of the ICPEE 2021—2021 1st International Conference on Power Electronics and Energy, Bhubaneswar, India, 2 January 2021. [[CrossRef](#)]
10. Karnavas, Y.L.; Topalidis, A.S.; Drakaki, M. Development and Implementation of a Low Cost MCbased Brushless DC Motor Sensorless Controller: A Practical Analysis of Hardware and Software Aspects. *Electronics* **2019**, *8*, 1456. [[CrossRef](#)]

11. Kim, K.-H.; Youn, M.-J. DSP-Based High-Speed Sensorless Control for a Brushless DC Motor Using a DC Link Voltage Control. *Electr. Power Compon. Syst.* **2002**, *30*, 889–906. [[CrossRef](#)]
12. Chen, W.; Xia, C.; Xue, M. A Torque Ripple Suppression Circuit for Brushless Motors Based on Power DC/DC Converters. In Proceedings of the 3rd IEEE Conference on Industrial Electronics and Applications, Singapore, 3–5 June 2008; pp. 1453–1457. [[CrossRef](#)]
13. Sathyan, A.; Milivojevic, N.; Lee, Y.J.; Krishnamurthy, M.; Emadi, A. An FPGA-Based Novel Digital PWM Control Scheme for BLDC Motor Drives. *IEEE Trans. Ind. Electron.* **2009**, *56*, 3040–3049. [[CrossRef](#)]
14. Vindel, D.; Haghbin, S.; Rabiei, A.; Carlson, O.; Ghorbani, R. Field-Oriented Control of a PMSM Drive System Using the DSPACE Controller. In Proceedings of the 2012 IEEE International Electric Vehicle Conference, Greenville, SC, USA, 4–8 March 2012. [[CrossRef](#)]
15. Milik, A.; Rudnicki, T. A Mixed Hardware-Software Implementation of a High-Performance PMSM Controller. *Electronics* **2023**, *12*, 440. [[CrossRef](#)]
16. Yao, G.; Yang, Y.; Wang, Z.; Xiao, Y. Permanent Magnet Synchronous Motor Control Based on Phase Current Reconstruction. *Electronics* **2023**, *12*, 1624. [[CrossRef](#)]
17. Bida, V.; Samokhvalov, D.; Al-Mahturi, F. PMSM Vector Control Techniques—A Survey. In Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Moscow and St. Petersburg, Russia, 29 January–1 February 2018. [[CrossRef](#)]
18. Sakunthala, S.; Kiranmayi, R.; Mandadi, P.N. A Review on Speed Control of Permanent Magnet Synchronous Motor Drive Using Different Control Techniques. In Proceedings of the 2018 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), Chennai, India, 22–23 February 2018; pp. 97–102. [[CrossRef](#)]
19. Sumega, M.; Rafajdus, P.; Stulrajter, M. Current Harmonics Controller for Reduction of Acoustic Noise, Vibrations and Torque Ripple Caused by Cogging Torque in PM Motors under FOC Operation. *Energies* **2020**, *13*, 2534. [[CrossRef](#)]
20. Pindoriya, R.M.; Gautam, G.; Rajpurohit, B.S. A Novel Application of Pseudorandom Based Technique for Acoustic Noise and Vibration Reduction of PMSM Drive. In Proceedings of the IEEE Industry Applications Society Annual Meeting, Baltimore, MD, USA, 29 September–3 October 2019; pp. 1–8. [[CrossRef](#)]
21. Coballes-Pantoja, J.; Gómez-Fuentes, R.; Noriega, J.R.; García-Delgado, L.A. Parallel Loop Control for Torque and Angular Velocity of BLDC Motors with DTC Commutation. *Electronics* **2020**, *9*, 279. [[CrossRef](#)]
22. Sitharthan, R.; Krishnamoorthy, S.; Sanjeevikumar, P.; Holm-Nielsen, J.B.; Raja Singh, R.; Rajesh, M. Torque Ripple Minimization of PMSM Using an Adaptive Elman Neural Network-Controlled Feedback Linearization-Based Direct Torque Control Strategy. *Int. Trans. Electr. Energy Syst.* **2021**, *31*, e12685. [[CrossRef](#)]
23. Mercorelli, P. Control of Permanent Magnet Synchronous Motors for Track Applications. *Electronics* **2023**, *12*, 3285. [[CrossRef](#)]
24. Mishra, P.; Banerjee, A.; Ghosh, M. FPGA-Based Real-Time Implementation of Quadral-Duty Digital-PWM-Controlled Permanent Magnet BLDC Drive. *IEEE/ASME Trans. Mechatron.* **2020**, *25*, 1456–1467. [[CrossRef](#)]
25. He, C.; Xu, S.; Yan, B.; Wang, Z.; Wang, M. A Fixed-Point Position Observation Algorithm and System-on-Chip Design Suitable for Sensorless Control of High-Speed Permanent Magnet Synchronous Motor. *Electronics* **2023**, *12*, 3160. [[CrossRef](#)]
26. Prusak, D.; Karpel, G.; Kułakowski, K. The Architecture of a Real-Time Control System for Heating Energy Management in the Intelligent Building. *Energies* **2021**, *14*, 5402. [[CrossRef](#)]
27. Calleja, C.; Lopez-de-Heredia, A.; Gaztanaga, H.; Nieva, T.; Aldasoro, L. Validation of a Modified Direct-Self Control Strategy for PMSM in Railway-Traction Applications. *IEEE Trans. Ind. Electron.* **2016**, *63*, 5143–5155. [[CrossRef](#)]
28. Abassi, M.; Khlaief, A.; Saadaoui, O.; Chaari, A.; Boussak, M. Performance Analysis of FOC and DTC for PMSM Drives Using SVPWM Technique. In Proceedings of the 2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Monastir, Tunisia, 21–23 December 2015; pp. 228–233. [[CrossRef](#)]
29. Żelechowski, M. *Space Vector Modulated—Direct Torque Controlled (DTC—SVM) Inverter—Fed Induction Motor Drive*; Warsaw University of Technology: Warsaw, Poland, 2005.
30. Rudnicki, T.; Czerwiński, R.; Fręchowicz, A. Control Drivers for Pmsm Motor. *Zesz. Probl.—Masz. Elektr.* **2011**, *90*, 51–55.
31. Mokhtar, A.S.N.; Reaz, M.B.I.; Marufuzzaman, M.; Ali, M.A.M. Hardware Implementation of A High Speed Inverse Park Transformation Using CORDIC and PLL for FOC Brushless Servo Drive. *Electron. Electr. Eng.* **2013**, *19*, 23–26. [[CrossRef](#)]
32. Marufuzzaman, M.; Reaz, M.B.I.; Rahman, M.S.; Ali, M.A.M. Hardware Prototyping of an Intelligent Current Dq PI Controller for FOC PMSM Drive. In Proceedings of the International Conference on Electrical & Computer Engineering (ICECE 2010), Dhaka, Bangladesh, 18–20 December 2010; pp. 86–88. [[CrossRef](#)]
33. Vaclavek, P.; Blaha, P. Field Weakening in PMSM Model Based Predictive Control. In Proceedings of the 2010 IEEE International Conference on Power and Energy, Kuala Lumpur, Malaysia, 29 November–1 December 2010; pp. 330–335.
34. Fajkus, P.; Klima, B.; Hutak, P. High Speed Range Field Oriented Control for Permanent Magnet Synchronous Motor. In Proceedings of the International Symposium on Power Electronics Power Electronics, Electrical Drives, Automation and Motion, Sorrento, Italy, 20–22 June 2012; pp. 225–230. [[CrossRef](#)]
35. Graf, M.; Buchta, L.; Pohl, L. Nonlinear Predictive Controller Design of PMSM with Field Weakening Performance. In Proceedings of the IECON 2013—39th Annual Conference of the IEEE Industrial Electronics Society, Vienna, Austria, 10–13 November 2013; pp. 3001–3005. [[CrossRef](#)]
36. Mynar, Z.; Vesely, L.; Vaclavek, P. PMSM Model Predictive Control With Field-Weakening Implementation. *IEEE Trans. Ind. Electron.* **2016**, *63*, 5156–5166. [[CrossRef](#)]

37. Azcue-Puma, J.L.; Sguarezi Filho, A.J.; Ruppert, E. Direct-FOC with Fuzzy Current Control for Asynchronous Machine. In Proceedings of the 2013 IEEE International Conference on Industrial Technology (ICIT), Cape Town, South Africa, 25–28 February 2013; pp. 307–312. [[CrossRef](#)]
38. TIA/EIA-485-A; Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems. ANSI: Arlington, TX, USA, 1998.
39. T-REC-V.11; Electrical Characteristics for Balanced Double-Current Interchange Circuits Operating at Data Signalling Rates Up to 10 Mbit/s. ITU-T: Geneva, Switzerland, 1996.
40. Koczkodaj, W.W.; Orłowski, M. An orthogonal basis for computing a consistent approximation to a pairwise comparisons matrix. *Comput. Math. Appl.* **1997**, *34*, 41–47. [[CrossRef](#)]
41. Janicki, R.; Koczkodaj, W.W. A weak order approach to group ranking. *Comput. Math. Appl.* **1996**, *32*, 51–59. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.