Marcin PAPROCKI [1], Andrzej WAWRZAK [1], Krystian ERWIŃSKI [1], Kazimierz KARWOWSKI [1], Marek KŁOSOWIAK [2]
[1] NICOLAUS COPERNICUS UNIVERSITY, INSTITUTE OF PHYSICS
5 Grudziądzka St., 87-100 Torun, Poland
[2] INDUSTRIAL RESEARCH INSTITUTE FOR AUTOMATION AND MEASUREMENTS, DEPARTMENT IN TORUN (PIAP-OBRUSN)
107 Stefana Batorego St., 87-100 Torun, Poland

PC-based CNC machine control system with LinuxCNC software

Abstract

In the article a PC-based numerical machine control system is presented, communicating over EtherCAT bus with servodrives and auxiliary I/O devices. LinuxRTAI real-time operating system was implemented on the PC controller along with the LinuxCNC control software. EtherCAT software communication module was developed and integrated with LinuxCNC. The software module implements CANOpen (CIA301) application layer standard along with device profiles for servodrives (CIA402) and I/O devices (CIA401). This allows for fast bidirectional communication between the PC controller and servodrives or I/O devices. The CNC control system developed by the authors has simple construction and is very flexible. It can be easily adapted to machines of different configuration.

Keywords: CNC control system, open control system, realtime operating system, Linux RTAI, LinuxCNC, EtherCAT.

1. Introduction

CNC machine control systems can be divided into closed systems dedicated to particular machine types and open systems which are often based on industrial PC's (IPC) with CNC control software [1]. Open PC-based automation control systems can be easily adapted to machines of different configurations. These kinds of systems are becoming increasingly more popular among machine manufacturers.

Different kinds of communication interfaces between the controller and servodrives are used. Often these are simple stepper drive digital interfaces (CLK/DIR/EN). Most advanced IPC based systems utilize some form of Industrial Ethernet to interface servodrives and I/O devices. One popular solution is Beckhoff Automation TwinCAT CNC using EtherCAT [2] variant of Industrial Ethernet. EtherCAT is defined by IEC 61158 norm. EtherCAT datagrams are encapsulated in standard Ethernet frames (IEEE 802.3). Because of this standard Ethernet cables can be used to connect servodrives and I/O modules to the PC using a standard network interface card (NIC). The application layer uses the CANOpen [3] protocol with different device profiles depending on the devices used. EtherCAT is an excellent protocol for real-time control of servodrives. Communication with each drive and I/O device is handled by a single Ethernet frame cyclically transmitted by the master with data updated on the fly by each slave. There are many drives on the market that support CANOpen EtherCAT communication.

One solution for numerical control systems is LinuxCNC [4] implemented on a IPC with a real-time operating system and EtherCAT communication bus. This solution has large capabilities for controlling machines of different configurations. Using open control systems based on IPC's and EtherCAT is an excellent alternative to closed systems due to low cost and large programming capabilities.

2. Control system architecture

The prestented control system consists of a PC computer (with real-time Linux, LinuxCNC control software and EtherCAT communication stack implemented), servodrives, auxiliary devices such as I/O modules or variable feed drives for spindle control. All components of the control systems communicate with each other using EtherCAT bus with CANOpen application layer. Figure 1 presents the block schematic of the control system.

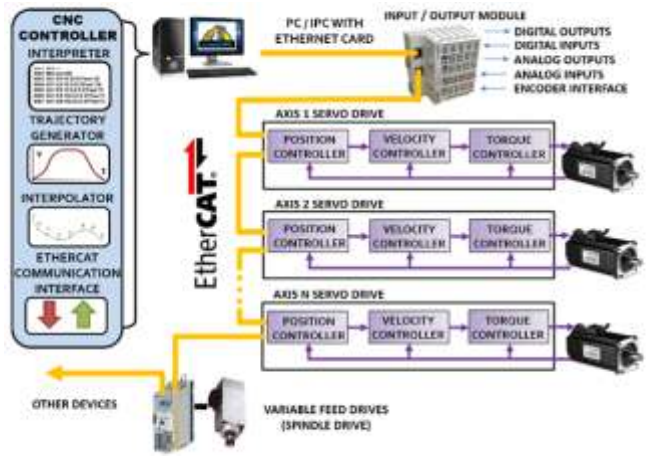

Fig. 1. Block schematic of the CNC control system

The CNC controller is a typical desktop PC with Intel Core i3 dual-core processor. Most kinds of modern PC's can be used but those with Intel Core i3, i5 or i7 series multi-core processors with SSD hard-drives achieve by far the best performance. As a standard, the LinuxCNC software can control up to 9 numerically controlled Cartesian axes. If necessary, the number of physically controlled joints (individual motors) can be much larger by using custom components and non-trivial kinematic transformations. The number of controlled I/O devices can be very large, even in the hundreds. In most CNC applications, all slave devices (drives, I/O's, VFD's) are handled by a single Ethernet frame with EtherCAT datagrams for each device embedded in the frame's payload. The number of handled devices is limited by the EtherCAT communication cycle imposed by the controller and the payload size of the Ethernet frame (around 1500 bytes). In the presented control system, the PC controller communicates with 4 servo drives and an I/O module cyclically every 1 ms. Devices from different manufacturers can be used provided they support EtherCAT communication conformant with the CANOpen application layer protocol. The presented system uses Delta Electronics ASD-A2-E servodrives and Beckhoff Automation I/O modules.

3. CNC control software

LinuxCNC control software is implemented on the PC-based CNC master controller. The software operates under LinuxRTAI (Real-Time Application Interface) real-time operating system [5], which enables deterministic execution of time-critical tasks. These are mainly motion control and real-time communication via EtherCAT. LinuxRTAI is based on a patched version of the standard Linux kernel using the microkernel approach. The real-time microkernel handles all real-time tasks and has full control over hardware interrupts. Since CNC control requires periodic execution of the motion control algorithms and communication, rate monotonic priority scheduling approach is used for the real-time tasks. Priorities are assigned based on task execution period with the shortest period task having highest priority. Task priorities are static which means that the higher priority task always preempts lower priority task and priorities cannot be changed while the tasks are running. All non-real time tasks of the standard Linux kernel have priorities lower than real-time tasks which effectively makes them run in background when real-time operation is in effect. Priorities also apply to hardware interrupts so that non-real time drivers cannot interfere with real-time operation. In order to improve performance, real-time and non-real-time tasks can be assigned to different processor cores. Figure 2 presents a block diagram of real-time and non-real time modules executed by the CNC controller.

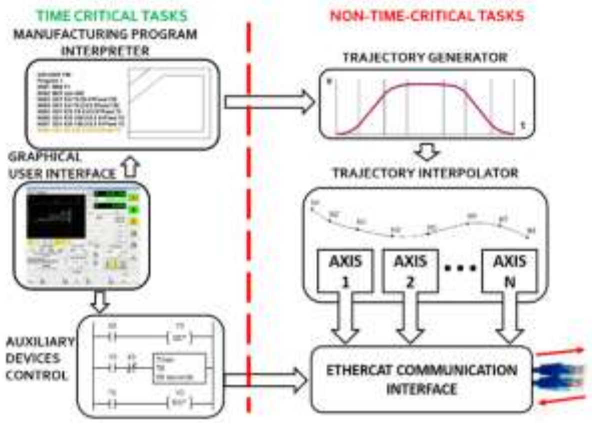

Fig. 2. CNC software modules divided into real-time and non-real time tasks

Figure 3 presents the structure of LinuxCNC software modules. LinuxCNC is divided into user-space Application Layer and the Hardware Abstraction Layer (HAL). The Application Layer consists of the main program which loads all other real-time and non-real time modules, handles configuration files, graphical user interface (GUI), G-Code interpreter [6] and I/O control. The user can choose from several available user space programs such as different GUI's, software oscilloscope or software PLC. Custom user-space applications can also be created in C, C++ or Python such as GUI's or program preprocessors. All of these software modules run as standard Linux tasks and are therefore non-real time tasks.

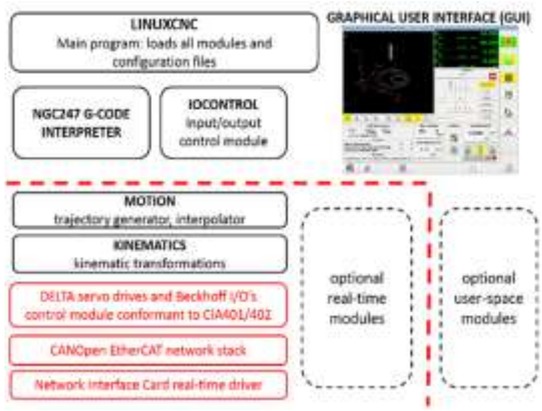

Fig. 3. Structure of LinuxCNC software blocks

The HAL includes real-time modules operated by the LinuxRTAI microkernel. Standard HAL modules loaded by LinuxCNC include the main motion control module (Motion) which includes the trajectory generator with trapezoidal federate profiling and interpolator. Different kinematic transformation modules can be used such as Cartesian or Tripod depending on the machine's configuration. In order to connect LinuxCNC with slave devices (servodrives and I/O modules) custom HAL modules had to be implemented by the authors. These included the real-time EtherCAT driver of the network interface card, EtherCAT stack conformant with the CANOpen application layer standard and device profile module. The device profile module controls Delta Electronics ASD-A2-E servodrives according to the CIA402 standard and Beckhoff I/O module according to the CIA401 standard. Other custom real-time modules can be implemented in the HAL. All modules can share data with each other using shared memory. Shared memory blocks as well as FIFO's can also be used to exchange data between user space applications and HAL modules.

LinuxCNC implements a wide variety of G-Code instructions used to program the multiaxis machine. These include technological cycles such as drilling or pocket machining and program flow instructions (conditional statements, loops). Using G-code the user can also control the look-ahead functionality which is part of the trajectory generation. G61 code forces the machine to reach every defined point. G64 implements look-ahead toolpath smoothing with optional tolerance which greatly improves program execution speed and improves machining accuracy especially for toolpaths comprised of short line segments.

4. Ethercat communication bus in the CNC control sytem

Standard Ethernet used with the TCP/IP [7] protocol does not allow sufficient time determinism required for synchronous control of a CNC machine's servo drives. EtherCAT protocol performs isochronous communication fundamental for motion control applications. The physical layer of EtherCAT is identical to Ethernet. This enables the use of standard Network Interface Cards in the master PC. The physical topology of the EtherCAT network is usually linear with a single master while implementing a logical link using full-duplex transmission. Usually a single Ethernet frame is sent periodically by the master (PC-based CNC controller) which contains all data intended for every slave. The frame has also empty data fields reserved for data returned by each slave. If data size exceeds the frame's payload several frames are sent sequentially. When a slave device receives the frame, data is read and written on the fly and the updated frame is forwarded to the next slave. The last slave returns the updated frame to the master. Because hardware frame processing is used by the slaves propagation delays induced by each slave is minimal (between 230 ns and 1 μs). EtherCAT uses a distributed clock synchronization mechanism in order to compensate for propagation delay and cycle time jitter introduced by the PC-based master. EtherCAT communication is therefore highly deterministic and short cycle times are possible even when the master is PC-based and the network consists of many different devices. The dataflow schematic of EtherCAT communication is shown in Figure 4.

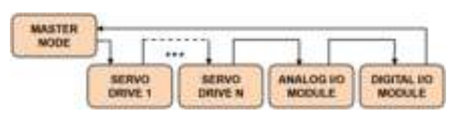

Fig. 4. Dataflow in EtherCAT network

EtherCAT application layer is based on the CANOpen protocol. The CANOpen application layer, defined in CIA301 standard is built around the Object Dictionary (OBD). The Object Dictionary is a data structure containing all communication parameters and process data exchanged between devices in the network. Device profiles are used to define a standard OBD structure for each device type. Device profiles for I/O modules and servodrives are defined by CIA401 and CIA402 standards respectively. CIA402 defines standard addresses for position demand, feedforward values, controller gains, I/O state, drive status, following error, operating mode and other typically used variables. It also defines operating modes such as homing, cyclic or profiled position, velocity and torque mode. Because EtherCAT is based upon standard Ethernet it allows sending standard TCP/IP messages when the bus isn't utilized by EtherCAT transmissions.

5. ASD-A2-0721-E servo drives, controller parameter tuning

The developed control system utilize Delta Electronics ASD-A2-0721-E servodrives. The basic drive parameters are: power 750 W, rated torque 2.4 Nm, rated speed 3000 rev/min, position measurement resolution 1280000 inc/rev.

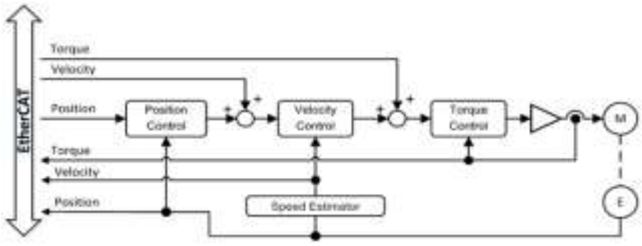

Fig. 5. ASDA-A2-0721-E servo drive controller structure in Cyclic Synchronous Position Mode

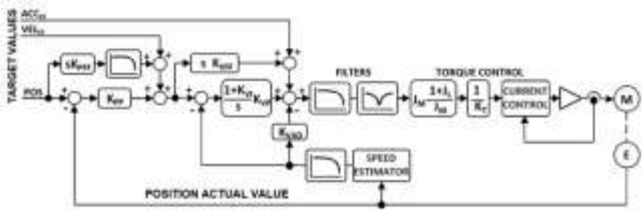

Fig. 6. Position and velocity controller structure of ASD-A2-0721-E servo drive

The control system uses CIA402 Cyclic Synchronous Position Mode for servo drive communication. In this mode position demands and velocity and acceleration feedforward values are transmitted at every 1 ms communication cycle. Each servodrive responds with actual position, actual velocity and actual torque producing current (Fig 5). The drive control structure is presented in Figure 6 and is comprised of P position controller, PDFF (pseudo derivative feedback with feed-forward) velocity controller. Torque and velocity feedforward values from the CNC controller are added to the torque and velocity demand values. The torque controller is factory defined and is inaccessible by the user.

ASD-A2-0721-E controller gains were tuned using ASDA-Soft drive configuration software provided by the manufacturer. Two sets of controller gains were computed:

- First set was computed using default autotuning implemented in ASDA-Soft, achieved velocity controller bandwidth was 68 Hz,
- Second set was computed based on frequency analysis performed on the drive and using internal anti-resonance notch filters. Using this method, the velocity controller achieved 160 Hz bandwidth with 14 dB gain margin and 55^{o} phase margin [8].

6. Achieved drive parameters, example application

The presented control system consisted of an Intel Core i3 PC with a Realtek 8111/8168 NIC, four numerically controlled axes with ASD-A2-0721-E EtherCAT servodrives and Beckhoff I/O modules (EK1828 EtherCAT coupler, EL1008 digital inputs, EL2008 digital outputs and EL5101-0010 incremental encoder interface). Tests were performed using an experimental setup with a linear motion module (Fig. 7) and a 3-axis machine (Fig. 8) developed at PIAP-OBRUSN Torun. The linear motion module was equipped with ESSA linear encoder with 0.1 µm resolution and a Kubler incremental encoder with a resolution of 144000 inc/rev placed at the end of the ballscrew. Ballscrew pitch was 5 mm/rev. The motor connected to the ASD-A2-0721-E servo drive was equipped with a Hiperface encoder with a resolution of 1280000 inc/rev. Basic parameters of the system were presented in Table 1.

Fig. 7. Linear motion module

Fig. 8. 3-axis machine with the developed CNC control system

Tab. 1. CNC control system with ASDA-A2-0721-E servo drives

No.	Test	Parameters achieved	Details
1	Motor position measurement fluctuations for a stationary motor measured using the Hiperface encoder in the motor and the incremental encoder.	$\pm 2\times 2\pi/1280000$ no position change from the incremental encoder	Actual position equal to the demanded position. No movement.
2	Measurement of small displacement equal to: $+1\times 2\pi/144000$ rad and $-1\times 2\pi/144000$ rad (single increment of the incremental encoder in both directions).	$+1\times 2\pi/144000$ rad, $-1\times 2\pi/144000$ rad	Movement performed in both directions. Demanded rotary displacements were performed and were equal to ± 0.03472 µm of linear displacement for a 5 mm/rev ballscrew pitch.
3	Measurement of small displacement equal to: $+1\times 2\pi/144000$ rad and $-1\times 2\pi/144000$ rad (single increment of the incremental encoder in both directions) with 1.8 Nm load torque on the motor shaft	$+1\times 2\pi/144000$ rad, $-1\times 2\pi/144000$ rad	Movement performed in both directions. Demanded rotary displacements were performed and were equal to ± 0.03472 µm of linear displacement for a 5 mm/rev ballscrew pitch.
4	Measurement of maximum position error due to momentary change in load torque from 0 to 1.8 Nm. No change in demanded position.	$103\times 2\pi/144000$ rad	Motor was stationary before loading. Position demand equal to actual position.
5	Following error measurement while performing movement along a test toolpath (Fig. 9) with motor rated speed (3000 rev/min, 15 m/min).	Maximum following error less than 0.1 mm	Measurement performed for a single axis with a 5 mm/rev ballscrew.

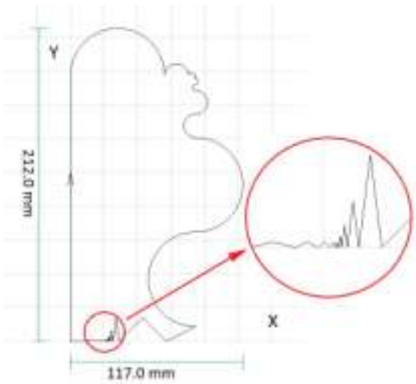

Fig. 9. Toolpath used to test following errors

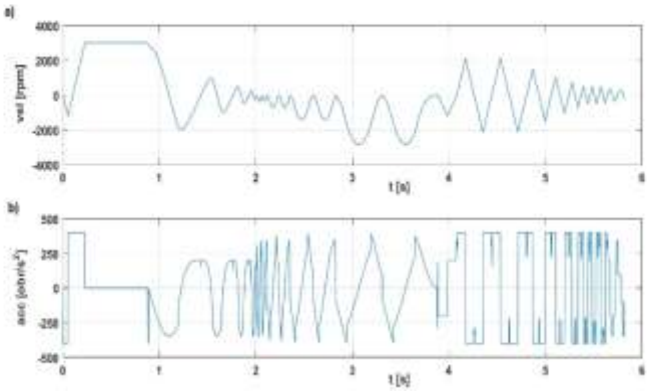

Fig. 10. Velocity (a) and acceleration (b) demand during execution of the test toolpath on the linear motion module

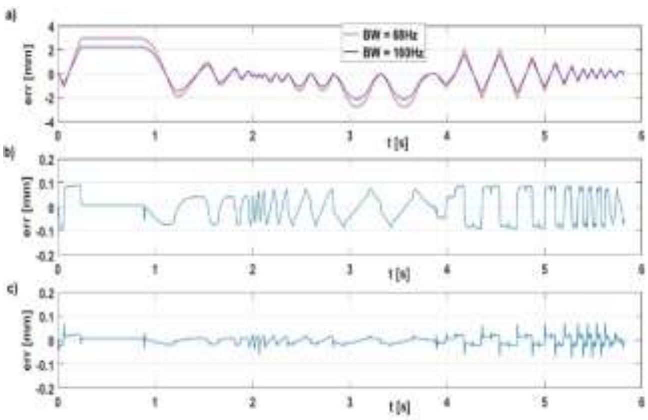

Fig. 11. Following error during execution of the test toolpath on the linear motion module.: a) without feedforward, b) feedforward – formula (1), c) feedforward – formula (2)

The toolpath used to test following error was presented in Figure 9. Feedrate and acceleration achieved during realization of the test toolpath were presented in Figure 10. The feedrate programmed in G-Code was 15000 mm/min which equaled to the rated motor speed with the ballscrew used. Following error measurement was performed on the linear motion module equipped with precise measurement devices. Test results which show servo drive following error while realizing the above toolpath were presented in Figure 11.

a) Without feedforward ($VEL_{FF} = 0, ACC_{FF} = 0$, Fig. 6), following errors were proportional to the velocity due to position P controller. At 160 Hz controller bandwidth and nominal speed the following error was 2.5 mm,
b) With velocity feedforward, proportional to velocity demand:

$$VEL_{FF} = K_1 \frac{d}{dt}(POS) \quad (1)$$

where: POS – position demand, factor K_1 was experimentally fine-tuned in order to achieve minimum following error. The following error was decreased below 0.1 mm. According to Figs 10b and 11b largest values of following errors are present during acceleration and deceleration.

c) With additional acceleration feedforward:

$$VEL_{FF} = K_1 \frac{d}{dt}(POS) + K_2 \frac{d^2}{dt^2}(POS) \quad (2)$$

Following error was further decreased. According to the presented figures largest following errors are present when the acceleration is changing (non-zero jerk). The negative side-effect are large current spikes during changes to acceleration.

7. Conclusions

The control system has simple construction, the components used are inexpensive and easily available. The LinuxCNC software enables execution of complex G-Code programs including toolpaths defined as polynomial splines (NURBS). The control system is easy to adapt to control machines of different configurations. EtherCAT communication is deterministic with very small cycle jitter. A single EtherCAT frame can handle all servo drives and I/O modules with propagation delays and jitter compensated by the servodrives. The CNC control system used with ASD-A2-E servodrives the was able to perform very small displacements (below 1 μm). It also achieved small following errors while performing movements with rated motor speeds. The presented CNC control system is a good alternative to existing solutions present on the market especially expensive closed CNC controllers.

8. References

[1] Huang H., Chi G. & Wang, Z.: Development and application of software for open and soft multi-axis EDM CNC systems, Int J Adv Manuf Technol, Vol. 86, No. 9 (2016): pp. 2689÷2700, DOI:10.1007/s00170-016-8353-6

[2] Chen Xin; Li Di; Wan Jiafu; et al.: A clock synchronization method for EtherCAT master, Microprocessors and Microsystems, Vol. 46, Part B, (2016): pp. 211÷218. DOI: 10.1016/j.micpro.2016.03.002

[3] López-López F., Gomis-Bellmunt O., Teixidó-Casas M., Rafecas-Sabaté J. and Muñoz-Gazquez J. P.: A novel educational platform to teach CANopen field bus. Comput. Appl. Eng. Educ., Vol. 19, No. 2, (2011): pp. 377÷384. DOI:10.1002/cae.20319

[4] Wawrzak A., Erwiński K., Karwowski K., Paprocki M., Kłosowiak M.: PC based CNC control system with EtherCAT fieldbus. Pomiary Automatyka Robotyka, R. 20, Nr 2/2016, 29-34, DOI: 10.14313/PAR_220/29.

[5] Erwinski K., Paprocki M. Grzesiak L. M., Karwowski K., Wawrzak A.: Application of ethernet powerlink for communication in a linux rtai open cnc system", IEEE Trans. on Industrial Electronics, vol. 60, no. 2, pp. 628-636, Feb. 2013. DOI: 10.1109/TIE.2012.2206348

[6] Ferreira J.C.E., Benavente J.C.T. & Inoue P.H.S.: A web-based CAD/CAPP/CAM system compliant with the STEP-NC standard to manufacture parts with general surfaces. J Braz. Soc. Mech. Sci. Eng., Vol. 39, No. 1, (2017): pp. 155÷176. DOI:10.1007/s40430-016-0528-4

[7] Mejías A, Herrera RS, Márquez MA, Calderón AJ, González I, Andújar JM.: Easy Handling of Sensors and Actuators over TCP/IP Networks by Open Source Hardware/Software. Sensors. Vol. 17(1), No. 94; (2017): pp. 1÷23. DOI:10.3390/s17010094.

[8] Aström K.J. and Hägglund T.,: PID Controllers: Theory, Design, and Tuning. 2nd ed., Research Triangle Park, NC: Instrum. Soc. Amer., 1995.

Received: 02.10.2016 *Paper reviewed* *Accepted: 02.12.2016*

Marcin PAPROCKI, PhD, eng.

MSc in Technical Physics in 2006 at Faculty of Physics Astronomy and Informatics, Nicolaus Copernicus University in Torun. PhD in Automation and Robotics in 2016 at the Faculty of Electrical Engineering, Warsaw University of Technology. His research interests include numerical machine control in particular trajectory error minimization using artificial intelligence.

e-mail: zwirek@fizyka.umk.pl

Andrzej WAWRZAK, MSc, eng.

MSc in Electronics in 1986 at the AGH University of Science and Technology in Krakow. His research interests include automation of technological processes, electrical drive control systems and numerical machine control systems.

e-mail: awawrzak@fizyka.umk.pl

Krystian ERWIŃSKI, PhD, eng.

MSc in Technical Physics in 2008 at Faculty of Physics Astronomy and Informatics, Nicolaus Copernicus University in Torun. PhD in Automation and Robotics in 2014 at the Faculty of Electrical Engineering, Warsaw University of Technology. His research interests include numerical machine control in particular time-optimal trajectory generation algorithms.

e-mail: erwin@fizyka.umk.pl

Kazimierz KARWOWSKI, PhD, eng.

BSc in Mechanical Engineering in 1970 at the University of Science and Technology in Bydgoszcz. MSc in Physics in 1975 at Poznan University. PhD in 1982 at Warsaw University of Technology. His research interests include CNC control systems, industrial automation and drive control systems.

e-mail: kkarwowski@fizyka.umk.pl

Marek KŁOSOWIAK, MSc, eng.

MSc in Electronics in 1981 at the Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology. His research interests include CNC control systems, CAD/CAM systems and technological process automation. Currently employed at PIAP-OBRUSN in Torun.

e-mail: m.klosowiak@obrusn.pl