**How to cite this article:**
Authors: Marcin Paprocki, Andrzej Wawrzak, Krystian Erwiński, Marek Kłosowiak
Title of article: „Elastyczny układ sterowania CNC maszyn bazujący na PC" ("Flexible PC-based CNC machine control system")
*Mechanik*, Vol. 91, No. 4 (2018): pages 299–303
DOI: https://doi.org/10.17814/mechanik.2018.4.46

# Flexible PC-based CNC machine control system

## Elastyczny układ sterowania CNC maszyn bazujący na PC

**MARCIN PAPROCKI**
**ANDRZEJ WAWRZAK**
**KRYSTIAN ERWIŃSKI**
**MAREK KŁOSOWIAK ***

**In the article a PC-based CNC machine control system is presented which communicates with servo drives and auxiliary input/output devices via EtherCAT bus. LinuxRTAI real-time operating system and LinuxCNC control software were implemented in the PC controller. A software module implementing EtherCAT communication with the servo drives was developed and integrated with LinuxCNC. Experimental results were presented which show the trajectory smoothing capabilities of the control system. Experimental results were also presented that show following errors obtained by running an example trajectory on a linear motion module. Basic parameters that show the control systems capabilities obtained during the experimental tests have been presented. The CNC control system presented in this article was implemented on a 3-axis machine.**

**KEYWORDS: CNC control system, open control system, real time operating system, Linux RTAI, LinuxCNC, EtherCAT**

CNC machine control systems are divided into closed control systems that prevent modification of the system architecture, and open systems, easy to configure, which can be adapted to different machines [1]. Open systems are often based on industrial computers (IPCs) with CNC software and are increasingly used to control multiple machines. Depending on the manufacturer, there are various communication buses in these systems, with one-way buses being the most popular – as for controlling stepped drives – sending CLK, DIR, ENABLE signals.

The advanced computer numerical control systems based on IPC computers are manufactured by Beckhoff Automation GmbH [2]. They are controlled from the built-in PC controller with integrated inputs and outputs (embedded PCs) or from an industrial multi-core PC. TwinCAT CNC [3] is used as the control program.

Communication between the IPC computer (constituting a CNC controller) and servo drives as well as input and output systems – both discrete and analog – is carried out via the EtherCAT bus [4] in accordance with IEC 61158. The physical and linear layers are compatible with the standard Ethernet frame IEEE 802.3. Thanks to this it is possible to connect servo-drives and input and output systems via typical Ethernet cables to the PC's port.

The CANopen communication protocol [5] was applied in the application layer of the bus stack. Many servodrives equipped with Ethernet ports with the CANopen communication protocol are offered on the market. The bus is perfectly suited for computer communication with real-time drives. Support for all servo drives, i.e. sending data and reading from servo drives and input/output systems, is carried out cyclically in one transmitted Ethernet frame.

One of the solutions of numerical control systems of machines are control systems based on IPC computers with LinuxCNC software [6], working in a Linux environment in real time and equipped with EtherCAT communication bus. It is a cheap solution, but with great adaptability to control variously configured machines.

Due to the low cost and high programming possibilities, control systems based on IPC computers and EtherCAT buses are a good alternative to closed systems.

### Control system architecture

In the subject control system, the master controller is a PC with the Linux RTAI (real time application interface) operating system and the LinuxCNC control program. The controller controls the servo drives of individual mechanical axes and auxiliary automation devices (such as frequency converter, inputs and outputs). All control system components communicate with each other via the EtherCAT bus. Fig. 1 shows the construction of the control system.

The control computer is a typical PC computer with an Intel Core i3 processor (dual core) with an Ethernet communication port. It is recommended to use PC or IPC computers with Intel Core i3, i5 or i7 processors and SSD disks. By default, nine numerical axes are planned, although there is a possibility of increasing their number. There can be a lot of entrance and exit (even several hundred). In one EtherCAT frame, up to 1500 B information is sent, which is used to operate servo drives, input and output systems and

---

* Dr inż. Marcin Paprocki (zwirek@fizyka.umk.pl), mgr inż. Andrzej Wawrzak (awawrzak@fizyka.umk.pl), dr inż. Krystian Erwiński (erwin@fizyka.umk.pl) – Instytut Fizyki, Uniwersytet Mikołaja Kopernika w Toruniu; mgr inż. Marek Kłosowiak (m.klosowiak@obrusn.pl) – Przemysłowy Instytut Automatyki i Pomiarów (PIAP) w Warszawie, Oddział Badawczo-Rozwojowy Urządzeń Sterowania Napędów w Toruniu

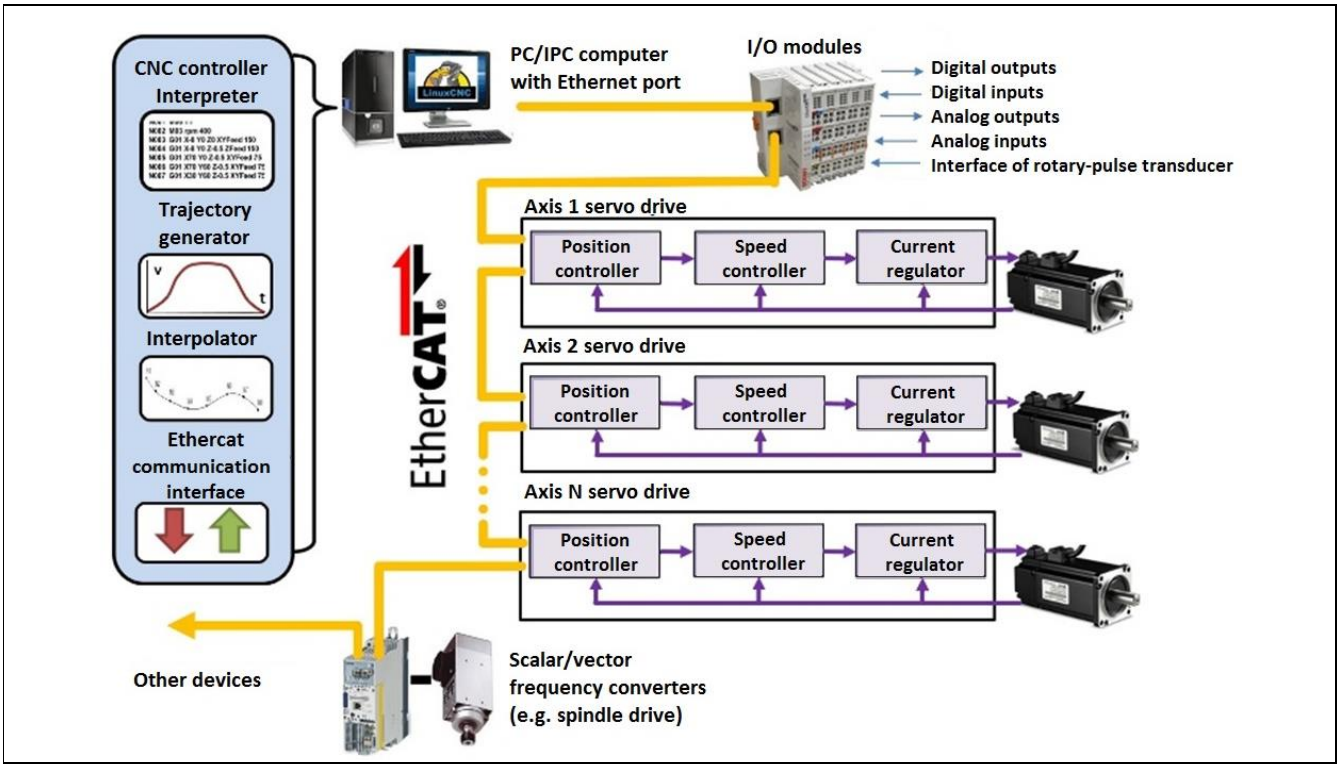

Fig. 1. Construction of the control system

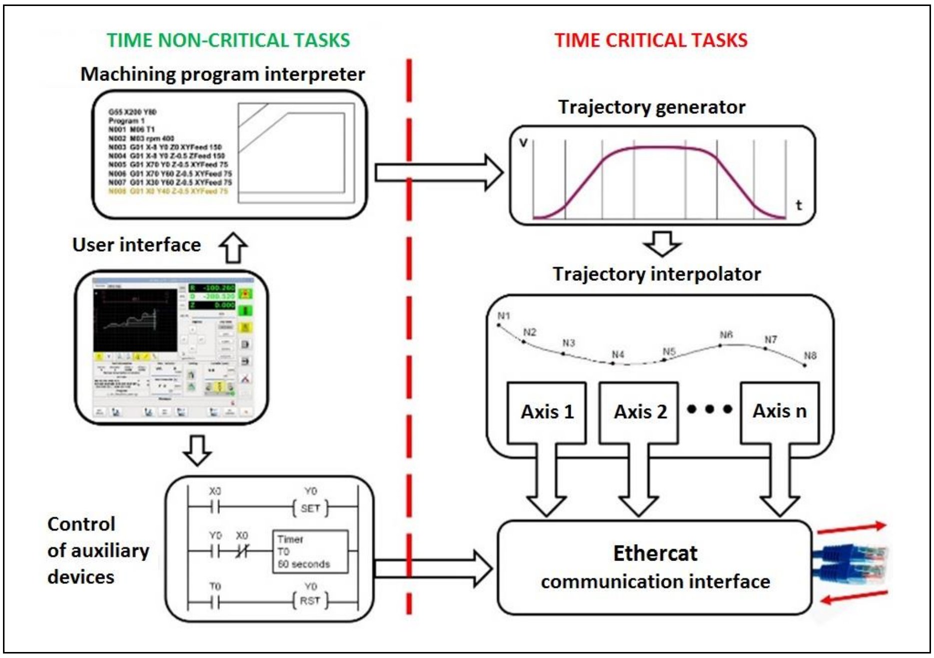

Fig. 2. Division of CNC program tasks into non-critical and deterministic tasks, carried out in strictly defined time periods

other drives. Axle servo drives do not have to come from a single manufacturer – it's important to have an Ethernet port with an EtherCAT communication stack and a CANopen communication protocol. As modules of discrete and analog input and output systems as well as modules for handling linear rulers (rotary-pulse transducers) Beckhoff products were used. In the developed control system successive EtherCAT communication frames are sent cyclically every 1 ms.

## CNC control program

The computer controlling the control system has been equipped with the LinuxCNC control program operating in the Linux RTAI environment [7], which enables execution of time-critical tasks in real time. Linux RTAI includes the standard Linux operating system kernel and real-time microkernel. The real-time microkernel accomplishes a small number of deterministic tasks of the CNC program and has direct access to hardware or clock interruptions. Real-time tasks have the highest priority. The Linux kernel performs other non-critical tasks temporarily, when none of the real-time tasks is performed. Fig. 2 presents modules of the LinuxCNC program, whose implementation is non-critical, and deterministic tasks, carried out cyclically in strictly defined time periods.

The LinuxCNC program is divided into the application part and the hardware abstraction layer (HAL). The application part includes: the master program loading the CNC program components together with configuration files, graphical user interface GUI (graphical user interface), G-program user code interpreter and machine input and auxiliary outputs module. It is possible to add your own modules to control the elements of the machine and to create your own GUI interfaces (or select them from several available in the LinuxCNC program). As part of the GUI, there is the option of displaying selected measurement results or using a virtual oscilloscope. The application part of the program is not time critical. In turn, the HAL hardware layer is time-critical and closely related to the Linux RTAI real-time micro-kernel. One can enter your own real-time modules in the HAL layer.

The developed control system in the LinuxCNC program implemented the EtherCAT bus communication stack and special modules for servo drives (servo drives from Delta Electronics, Inc.) and I / O, compatible with the CANopen protocol – communication profiles CiA402 and CiA401.

The LinuxCNC program implements a very extensive set of G-codes (RS274NGC) [8] for programming multi-axis CNC machines. The G61 code ensures accurate achievement of each point defined in the technology program. The G64 code performs the Look Ahead function, which is responsible for smoothing the trajectory (obtaining a continuous derivative of the movement path) and improving the quality of the machining. The LinuxCNC program allows for trapezoidal speed profiling.

## EtherCAT bus in the CNC control system

The standard Ethernet TCP/IP communication protocol [9] makes it impossible to obtain sufficient determinism ensuring synchronous servo control in CNC multi-axis machines. Unlike Ethernet TCP/IP, the EtherCAT protocol performs isochronous transmission and meets the high demands placed on traffic control applications.

In the physical layer, i.e. in layer 1 of the OSI (open systems interconnection) model, the EtherCAT protocol is identical to the analogous Ethernet TCP/IP protocol layer, which allows the use of standard network adapters available in PCs. In a typical EtherCAT network, there is one supervisor and a plurality of slave devices (maximum 65,535). The network has a logical ring structure. In the CNC control, the EtherCAT frame is sent by the master device, i.e. a PC, and contains data for each of the slave devices in the network. During the receipt and further forwarding of the frame, each slave implements the process of reading and modifying data and entering new data into the frame. The frame is forwarded by each subsequent network node and then goes back to the PC.

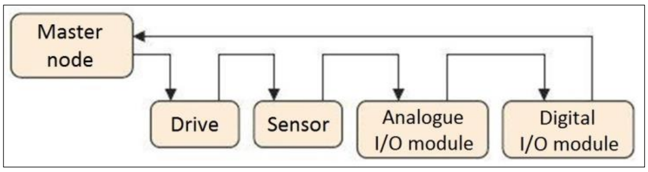

Fig. 3. Diagram of information flow in the EtherCAT bus

Frame propagation delays are minimal (from 230 ns to 1 μs for each network node). An exemplary diagram of information flow in the EtherCAT bus is shown in fig. 3.

The application layer of the EtherCAT stack is based on the CANopen communication protocol, which contains the OBD object library (CiA301). The structure of the object dictionary OBD contains data on all parameters relating to the EtherCAT communication stack as well as process variables sent from the supervisory device to the slave devices. Depending on the type of slave devices, there are several communication profiles (device profiles). They define a set of objects containing the configuration of process variables characteristic of a given node type. The communication profile of input and output systems is defined in the CiA401 standard, and servo drives in the CiA402 standard. Each servo drive that supports the EtherCAT protocol must have a library of OBD in the CiA402 standard. The objects defined by CiA402 can be information such as o: set position, actual position, velocity value, current value, logic state of digital inputs/outputs, values of GID regulator and feedforward control factor coefficients, servo operation mode configuration (in torque mode, set speed or position). The developed CNC controller supports the CiA402 standard.

Alternatively, EtherCAT allows sending some data using the TCP/IP protocol bypassing the EtherCAT stack.

## Technological possibilities of the control system, programming

The LinuxCNC software has many technological functions that are available from the G-Code RS274NGC language. When it is necessary to implement complex shapes, it is possible to define polynomial spline polynomials B-Spline or NURBS (G5, G5.1, G5.2, G5.3 codes) in the technological program. There are many predefined program cycles to perform typical technological operations (e.g. drilling – G81-83, reaming – G85-89, threading – G84). The language interpreter allows the use of program instructions, such as program loops, conditional instructions, program jumps or calling subroutines.

The functions of smoothing the given motion path (blending) are important. The technological program stored in the G code is usually in the form of linear segments or arcs. At the interface of segments, the trajectory is not smooth, which results in large changes in the torque of the drive motors and large changes in speed and vibration in the mechanical system. Currently, CAD/CAM software is commonly used to generate the technological code. The program created in this way often consists of many very short linear sections, on which the machine cannot always reach the maximum programmed feed rates. Program execution (G61 function) is slower due to stopping traffic at the end of each line segment. The G64 smoothing function eliminates this problem by introducing an arc between two adjacent segments. Stop at the end of the segment is not necessary, and the machining process is much faster. To properly generate a feedrate profile, it is necessary to analyze a certain number of code segments in advance (the so-called look-ahead).

Fig. 4 shows an example of a traffic path consisting of short linear sections. The movement was carried out with turned off and enabled smoothing. Speed ranges are shown in fig. 5.

The time to overcome the trajectory at a given feed speed of 15 m/min without smoothing was much longer (nearly 5 seconds) than with the smoothing enabled (about 1.5 seconds). In practice, the lack of smoothing not only extends the machining time, but also negatively affects the accuracy of the trajectory reproduction and causes vibrations of the supports and deterioration of the quality of the surface being processed.

Smoothing function – code G64–strives to smooth the set trajectory so that the feed speed is as close as possible to the set speed. For high feed speeds, this may result in significant discrepancies between the set and actually realized trajectory. It is possible to specify the acceptable deviation of the smoothed trajectory from the trajectory given with the command: code G64 Px (where x is the specified deviation in mm). The distance of the arc of the arc fitted at the interface of the linear segments from the common point of both segments is then equal to the set deviation.

By modifying this parameter, the user achieves a compromise between the speed of the implementation of the given trajectory and the accuracy of its reproduction.

Fig. 6 compares a fragment of a smoothed motion path using different contour error deviations. The maximum set deviations of the smoothed trajectory from the set trajectory were not exceeded.

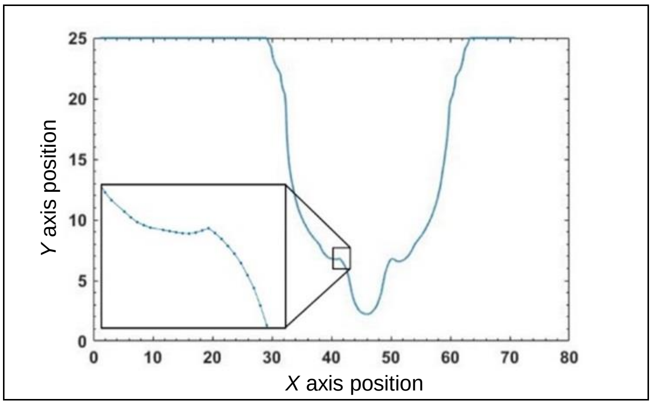

Fig. 4. Exemplary traffic path consisting of short linear sections

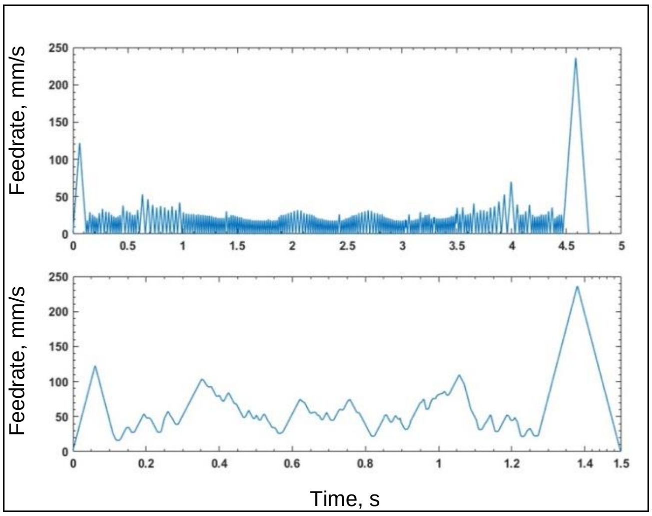

Fig. 5. Advances in feed speed of the axes with off (upper graph) and on (bottom graph) smoothing

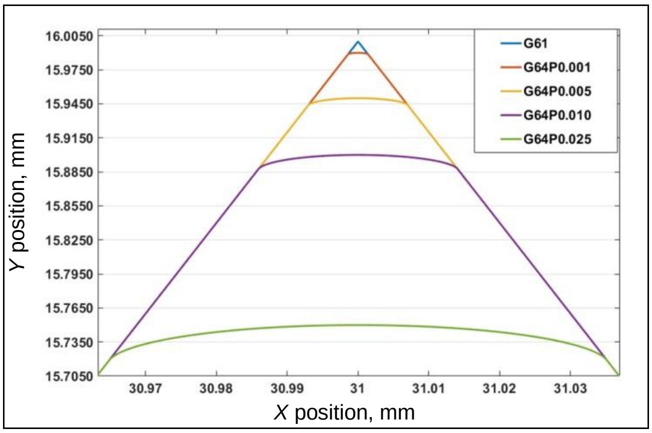

Fig. 6. Fragment of the trajectory (corner) with various contour error deviations

### Parameters achieved, exemplary implementation

The control system consisted of: a PC with an Intel Core i3 processor, a network card with an Ethernet port type Realtek 8111/8168B [10], four numerical axes with servo drives ASDA-A2-0721-E [11], an input and output coupling module Ethernet port EK1828 [12], discrete input systems EL1008 [13], discrete outputs EL2008 [14] and a module enabling the connection of the incremental rotary-impulse transmitter EL5101-0010 [15] by Beckhoff. Parameters of ASDA-A2-0721-E servo drives from Delta Electronics, Inc. the following were: power 750 W, nominal torque 2.4 Nm, nominal speed 3000 rev/min, resolution of measuring systems position 1280000 imp/rev. The ASDA-A2-0721-E servomotor position and speed controllers parameters were selected in the ASDA-Soft utility software. Thanks to the standard auto-tuning function, the frequency response of the speed regulator BW = 68 Hz was obtained.

**TABLE. CNC control system with ASDA-A2-0721-E drives**

| Description of the test | The parameters achieved | Remarks |
|---|---|---|
| The position fluctuation read from the Hyperface motor measuring transducer for the non-moving system. The position readout from the 144000 imp / rev rotational-pulse transducer did not change | ±2 × 2π/1 280 000 | The set position is equal to the actual engine position. No movement |
| Measurement of the realization of a given rotary motion with a value of:<br>• +1 × 2π/144 000 rad,<br>• –1 × 2π/144 000 rad<br>– i.e. 1 increment of rotary-pulse transducer in both directions | +1 × 2π/144 000 rad<br>–1 × 2π/144 000 rad | The movement was carried out in both directions. Displacements have been achieved:<br>+1 × 2π / 144000 rads and –1 × 2π / 144000 rad, i.e. ± 0.03472 μm; rolling screw with a stroke of 5 mm/rev |
| Measurement of the realization of a given rotary motion with a value of:<br>• +1 × 2π/144 000 rad,<br>• –1 × 2π/144 000 rad<br>– i.e. 1 increment of rotary-pulse transducer in both directions. The motor shaft was loaded with a torque of 1.8 Nm | | |
| Measurement of the momentary maximum position error at a step change of the motor load torque of 1.8 Nm (the set position did not change) | 103 × 2π/144 000 rad | Before loading, the actual position was equal to the position of the motor shaft |
| Measurement of lagging error in the implementation of the set trajectory (Fig. 9) at a preset speed of 3000 rpm (15 m / min), equal to the nominal speed of the motor | The maximum lag error does not exceed 0.1 mm | Measurement for one axis (ball screw with a stroke of 5 mm/rev) |

Due to the low BW value, the controller parameters were again selected using the frequency analysis function and the resonant frequency suppression filters. This time the bandwidth of the speed regulator BW = 160 Hz was obtained and 14 dB of the amplitude reserve and 55° of the phase reserve [16] were retained.The tests were carried out on a station consisting of a linear motion module (fig. 7) and a three-axis machine (Fig. 8) implemented in PIAP-OBRUSN in Toruń. The linear motion module was equipped with a 1 μm measuring scale and an incremental rotary-pulse transducer (Kubler) with a resolution of 144000 imp/rev. The pitch of the helical gearbox was 5 mm/rev. The parameters of the system are presented in the table. Fig. 9 shows the trajectory used to test lagging errors.

The speed and acceleration of the linear motion module during the test trajectory is shown in fig. 10. The feed of the working motions in the CNC file was 15000 mm/min, which corresponded to the nominal engine speed, i.e. 3000 rpm. Measurements of lagging errors were carried out on a linear motion module equipped with precise measuring systems.

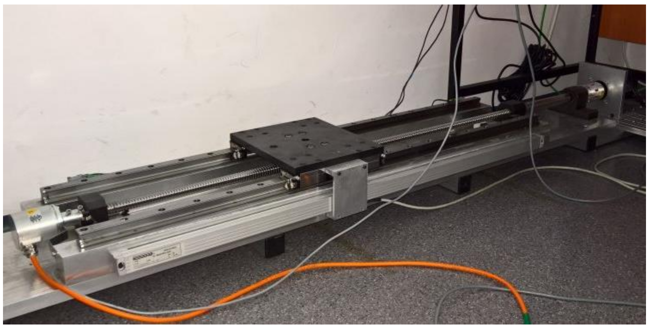

Fig. 7. Linear motion module

Fig. 8. Machine with a developed CNC control system

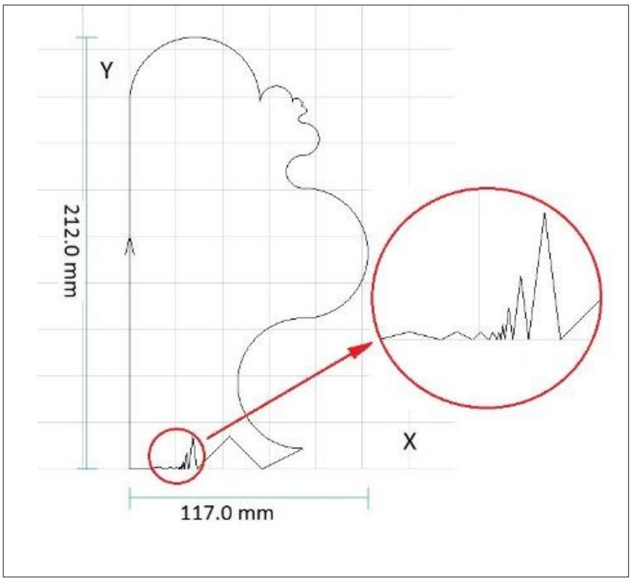

Fig. 9. Trajectory used to test lagging errors

The test results for the various leading values from velocity and acceleration $VEL_{FF}$ and $ACC_{FF}$ are shown in fig. 11.

When controlling without leading values from velocity and acceleration ($VEL_{FF} = 0$, $ACC_{FF} = 0$), lagging errors due to the position regulator type *P* were proportional to the current speed and at the servo drive bandwidth *BW* = 160 Hz and the nominal speed of the motor were about 2.5 mm.

When generating a leading signal proportional to the current speed:

$$VEL_{\mathrm{FF}} = K_1 \frac{\mathrm{d}}{\mathrm{d}t}(POS) \tag{1}$$

where: *POS* – set position, $K_1$ – factor chosen so as to minimize the lagging error for the nominal speed.

Reduction of the error of following the level below 0.1 mm was obtained. According to fig. 10b and fig. 11b, the highest error values occur during the servo acceleration/deceleration phase.

When generating a pre-signal including acceleration:

$$VEL_{\mathrm{FF}} = K_1 \frac{\mathrm{d}}{\mathrm{d}t}(POS) + K_2 \frac{\mathrm{d}^2}{\mathrm{d}t^2}(POS) \tag{2}$$

A further reduction in lagging error was obtained. The graph shows that the biggest errors occur during the "rush" phase of the servo drive. The side and unfavorable effect of this control was the appearance of significant current pulses (jerks) at step changes in acceleration.

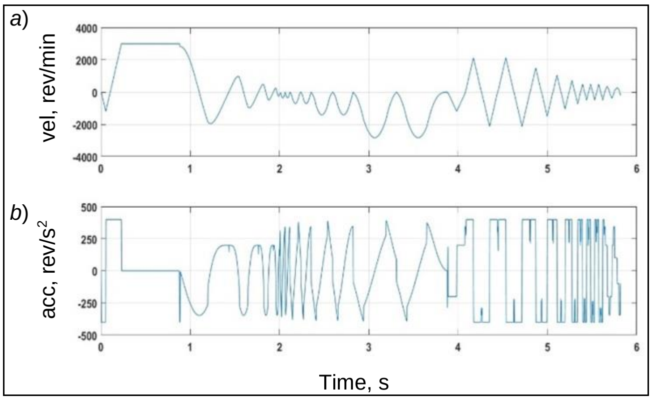

Fig. 10. Velocity (*a*) and acceleration (*b*) asked for the linear motion module during the test trajectory

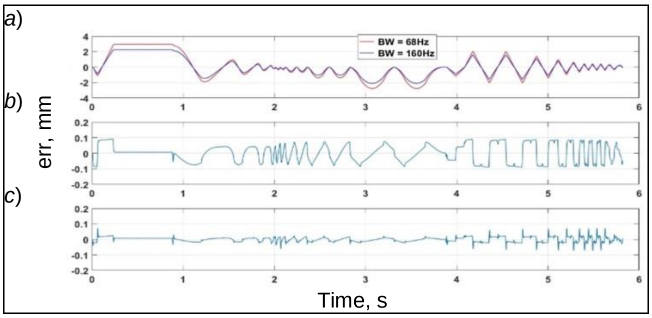

Fig. 11. Errors of following the linear motion module during the test trajectory: a) without feedforward ($VEL_{FF} = 0$, $ACC_{FF} = 0$), b) with feedforward–formula (1), c) with feedforward – pattern (2)

## Conclusions

The control system is characterized by a simple construction, and its components are cheap and available. The system software enables the creation of sophisticated application programs in G-codes as well as describing the trajectory using NURBS curves. The control system can easily be adapted to control machines with different configurations. Communication via the EtherCAT bus is deterministic, with small spreads in the cyclic transmission and reading of data from servo drives and input and output systems. One transmitted frame supports all servo drives, and its delay when transmitted by servo drives is very small and compensated in servo drives. The control system with the drives used was characterized by very high resolution. The displacements were carried out with an error below 1 µm. The lagging errors at speeds equal to the maximum machine axis speeds were also low.

The developed control system is a good alternative to commercially available CNC machine control systems, especially for expensive closed-loop control systems.

REFERENCES

1. Huang H., Chi G., Wang Z. „Development and application of software for open and soft multi-axis EDM CNC systems”. *International Journal of Advanced Manufacturing Technology*. 86, 9 (2016): pp. 2689–2700, DOI:10.1007/s00170-016-8353-6.
2. www.beckhoff.pl (11.03.2017).
3. www.beckhoff.com/english.asp?highlights/twincat-cnc/default.htm (11.03.2017).
4. Chen Xin, Li Di, Wan Jiafu et al. „A clock synchronization method for EtherCAT master”. *Microprocessors and Microsystems*. 46, Part B (2016): pp. 211–218, DOI: 10.1016/j.micpro.2016.03.002.
5. López-López F., Gomis-Bellmunt O., Teixidó-Casas M., Rafecas-Sabaté J., Muñoz-Gazquez J.P. „A novel educational platform to teach CANopen field bus”. *Computer Applications in Engineering Education*. 19, 2 (2011): pp. 377–384, DOI:10.1002/cae.20319.
6. Wawrzak A., Erwiński K., Karwowski K., Paprocki M., Kłosowiak M. „PC based CNC control system with EtherCAT fieldbus”. *Pomiary Automatyka Robotyka*. 20, 2 (2016): pp. 29–34, DOI: 10.14313/PAR_220/29.
7. Erwński K., Paprocki M., Grzesiak L.M., Karwowski K., Wawrzak A. “Application of Ethernet Powerlink for communication in a Linux RTAI open CNC system”. *IEEE Transactions on Industrial Electronics*. 60, 2 (2013): pp. 628–636, DOI: 10.1109/TIE.2012.2206348.
8. 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”. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*. 39, 1 (2017): pp. 155–176. DOI:10.1007/s40430-016-0528-4.
9. Mejías A., Herrera R.S., Márquez M.A., Calderón A.J., González I., Andújar J.M. “Easy handling of sensors and actuators over TCP/IP networks by open source hardware/software”. *Sensors*. 17(1), 94 (2017): pp. 1–23, DOI:10.3390/s17010094.
10. bothhand.info/pdf/spec-8111b(140).pdf (11.03.2017).
11. www.delta.com.tw/product/em/motion/motion_servo/download/manual/DELTA_IA-ASDA_A2_UM_EN_20141212.pdf (11.03.2017).
12. https://download.beckhoff.com/download/document/io/ethercat-terminals/ek18xxen.pdf (11.03.2017).
13. https://download.beckhoff.com/download/document/io/ethercat-terminals/el10xx_el11xxen.pdf (11.03.2017).
14. https://download.beckhoff.com/download/document/io/ethercat-terminals/EL20xx_EL2124en.pdf (11.03.2017).
15. https://download.beckhoff.com/download/document/io/ethercat-terminals/el5101en.pdf (11.03.2017).
16. Aström K.J., Hägglund T. “PID Controllers: Theory, Design, and Tuning”. 2nd ed. Instrument Society of America, 1995. ■

---

Translation of scientific articles, their computer composition and publishing them on the website www.mechanik.media.pl by original articles in Polish is a task financed from the funds of the Ministry of Science and Higher Education designated for dissemination of science.

Ministry of Science and Higher Education
Republic of Poland