TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

INNOVATION OF THE ASPHERICAL LENSES POLISHING MACHINE

Diploma thesis

Study programme:	N2612 – Electrical Engineering and Informatics
Study branch:	3906T001 – Mechatronics
Author:	**Igor Prikhodko**
Supervisor:	Ing. David Lindr, Ph.D.

Liberec 2015

Technical University of Liberec
Faculty of Mechatronics, Informatics and Interdisciplinary Studies
Academic year: **2014/2015**

DIPLOMA THESIS ASSIGNMENT

(PROJECT, ART WORK, ART PERFORMANCE)

First name and surname:	**Igor Prikhodko**
Study program:	**N2612 Electrical Engineering and Informatics**
Identification number:	**M14000281**
Specialization:	**Mechatronics**
Topic name:	**Innovation of the aspherical lenses polishing machine**
Assigning department:	**Institute of Mechatronics and Computer Engineering**

Rules for elaboration:

1. Meet the principle function of three-axis polishing machine.
2. Design the way for its modernization with respect to the requirements imposed on the functionality of the machine.
3. Implement designed solution into the control unit and test its functionality on the experimental stand.
4. Design the HMI interface that enables comfortable control of the machine.
5. Create the final documentation of the diploma thesis and among others insert the project of your algorithm into the electronic attachment.

Scope of graphic works: **In respect to the documentation needs**

Scope of work report
(scope of dissertation): **c. 40–50 pages**

Form of dissertation elaboration: **printed/electronical**

Language of dissertation elaboration: **English**

List of specialized literature:

[1] **Craig J.,J.: Introduction to Robotics Mechanics and Control, Third Edition, Prentice Hall, 2005.**

[2] **JOHN, Karl-Heinz; TIEGELKAMP, Michael. IEC 61131-3: Programming Industrial Automation Systems : Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making Aids . 2nd ed. New York : Springer, 2010. 390 s. ISBN 978-3-642-12014-5.**

[3] **SIMOTION Motion Control TO Path Interpolation, Function Manual, Edition 04/2014**

Tutor for dissertation: **Ing. David Lindr, Ph.D.**
Institute of Mechatronics and Computer Engineering

Dissertation Counsellor: **Ing. Gennadiy Golodnykh, Ph.D.**
Novosibirsk State Technical Univerzity, Russia

Date of dissertation assignment: **1 December 2014**
Date of dissertation submission: **15 May 2015**

V. Kopecký
prof. Ing. Václav Kopecký, CSc.
Dean

L.S.

Technická univerzita v Liberci - 14 -

Kolář
doc. Ing. Milan Kolář, CSc.
Department Manager

Liberec, dated: 10 October 2014

DECLARATION

I hereby certify that I have been informed the Act 121/2000, the Copyright Act of the Czech Republic, namely § 60 - Schoolwork, applies to my master thesis in full scope.

I acknowledge that the Technical University of Liberec (TUL) does not infringe my copyrights by using my master thesis for TUL's internal purposes.

I am aware of my obligation to inform TUL on having used or licensed to use my master thesis; in such a case, TUL may require compensation of costs spent on creating the work at up to their actual amount.

I have written my master thesis myself using literature listed therein and consulting it with my thesis supervisor and my tutor.

Concurrently I confirm that the printed version of my master thesis is coincident with an electronic version, inserted into the IS STAG.

Date: 14.05.2015

Signature:

ABSTRACT

The main project goal was to develop a software and hardware configuration for testing stand, which will be simulate work of three axis aspherical lenses polishing machine.

Whole work was divided into four steps. The first step was to learn the information about the topic in the field of spherical lenses and polishing machine theory. The second step was to obtain the relationship between coordinates of the desired point and angles of eccentric's rotation. In the step three, was written script for the simulation and visualization results, obtained in step two. As step four was created test stand and corresponded software for it.

KEYWORD

Automation, microcontroller, motion controller, drives, aspherical lens, polishing, path interpolation, virtual cam, SIMOTION SCOUT, PROFIBUS DP.

TABLE OF CONTENTS

List of used figures

INTRODUCTION

Automation of production and individual technological processes is the transfer of tracking and monitoring functions, previously performed by human, to the machines. All operations of the physical and mental work, which can be formalized, may be subject for automation. The term "automated" process, in contrast to the concept of "automatic" process, indicates the need for human involvement at some stage, in order to maintain control over the process.

Typically, the automation solution is the creation of automated process control system (APCS), which represents complete solution to automate the technical processes, systems operator control means archiving and accounting data on the process, and also includes the typical elements of automation: executive devices, control unit, sensors and so on.

Creating an automated polishing machine allows: accelerate individual stages of work, increase the safety of the process, increases application area and allow creating different programs and algorithms of polishing.

1 FORMULATION OF THE PROBLEM

1.1 ACTUALITY OF THEME

Currently, aspheric lenses are widely used in almost all optical instruments and devices (cameras, camcorders, projectors, glasses, astronomical technique). Using one aspherical surface can replace 2-3 spherical, which leads to a dramatic reduction in the number of parts of the optical system and, therefore, reduce weight and size of the optical system. Also aspherical lens produces a better image through better correction of aberrations (aberration characterize various types of violations of the rays passing through the optical system) optical systems. When light rays are passed through an uncorrected convex lens, the rays at the periphery are bent more than the ones passing near the optical center. Thus, the peripheral rays produce many focal points on the principal axis before the actual focus. This is called longitudinal or axial spherical aberration. The use of aspherical lenses can change the curvature of the surface, so that the focus is not lost on the edges of the lens [1]. A simple spherical lens is created by the rotation of the pitch circle about the axis of symmetry. Such lenses are widespread and have long been known (glass balls filled with water 3 century BC). Spherical lenses have a number of limitations on the application and cannot cope with all the tasks which facing optical devices with the increase of technological progress. For new tasks required new technology, this has become an aspherical lens. The surface of these lenses is created by the rotation of the other curves, which are different from the circle (parabola, ellipse, and hyperbola) [1] [2] [3].

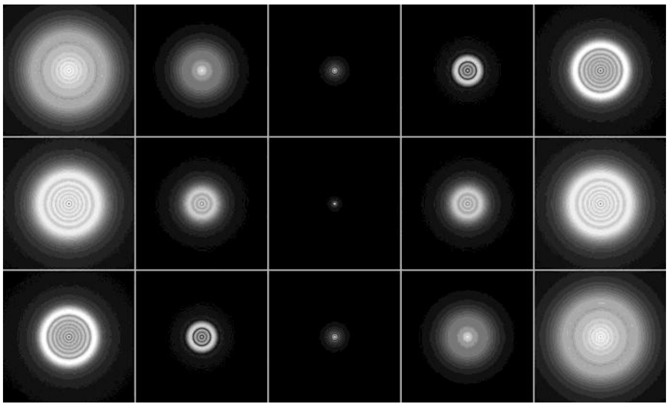


Fig. 1.1 A point source as imaged by a system with negative (top row), zero (middle row), and positive spherical aberration (bottom row). The middle column shows the focused image, columns to the left shows defocusing toward the inside, and columns to the right show defocusing toward the outside [1].

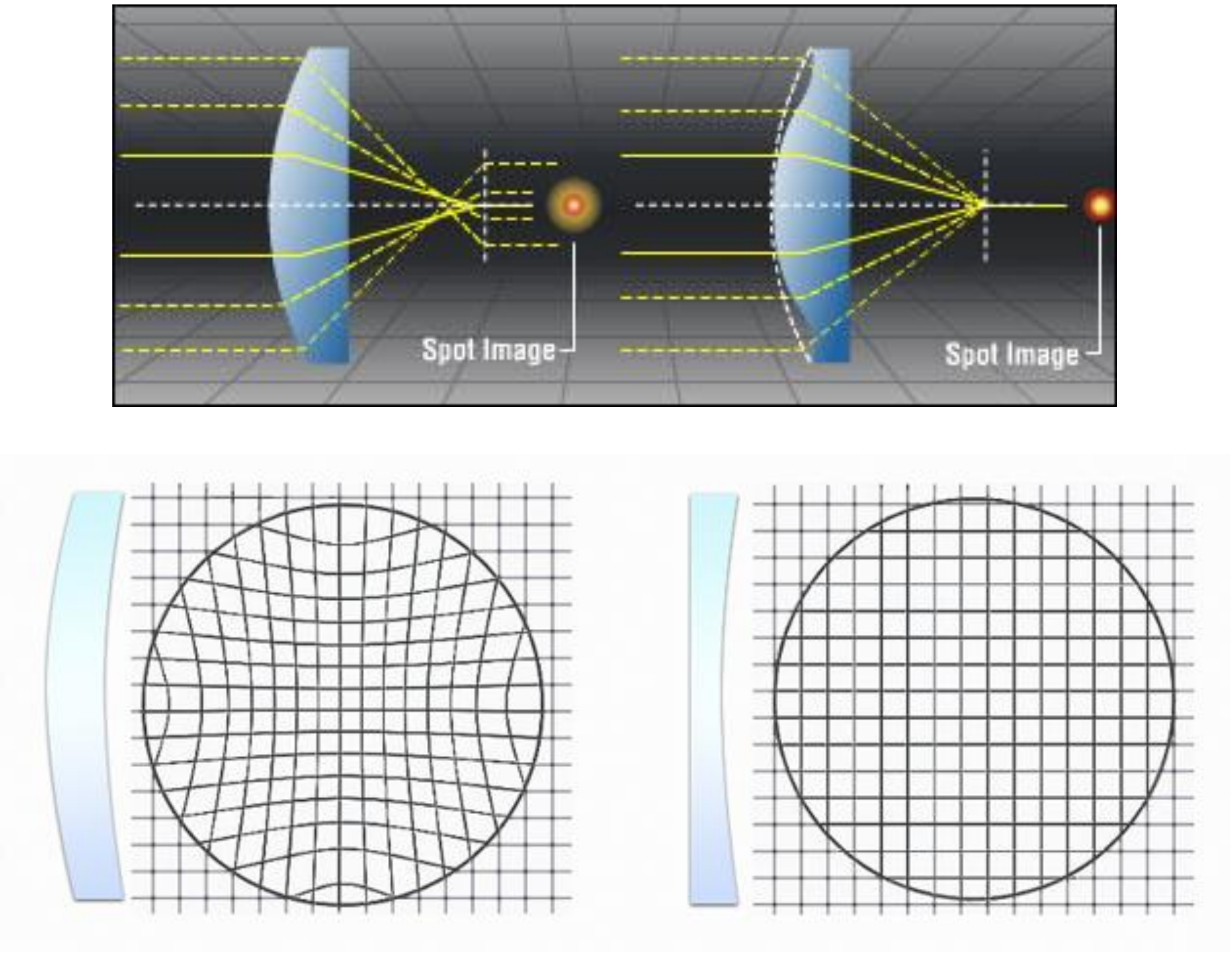


Fig. 1.2 Image formation spherical lenses (left) and an aspherical lens (right) [1]

Productions of aspheric lens technically are more complicated process and therefore more expensive then production spherical lenses. The using of aspherical lenses can improve the quality of the device in which they are used, but at the same time increase the total cost of the final product. Manufacturing of aspheric lenses require special equipment and is only partially identical with the production of spherical lenses (some steps in the production process of spherical and aspherical lenses are the same, therefore, can be perform on the same hardware) [3].

Polishing machines for spherical lenses technically simpler, cheaper to manufacture and more common. A shaping of aspheric lenses requires different equipment that makes such production more expensive and requires more manufacturing facilities and personnel who must be able to operate with all types of devices.

The market of polishing machines for spherical lenses is wide and full of competition; it creates dumping prices and improves the quality of the final product. The market offers a polishing machine at various levels, from the simplest, ending with hi-tech machines with lots of programmable modes and presets.

Market polishing machines for aspheric lenses has fewer proposals. Prices for these machines can be different by an order to the next higher. To work with such machines require specially trained staff. Often such machines can be used for spherical lenses (hybrid polishing machine).

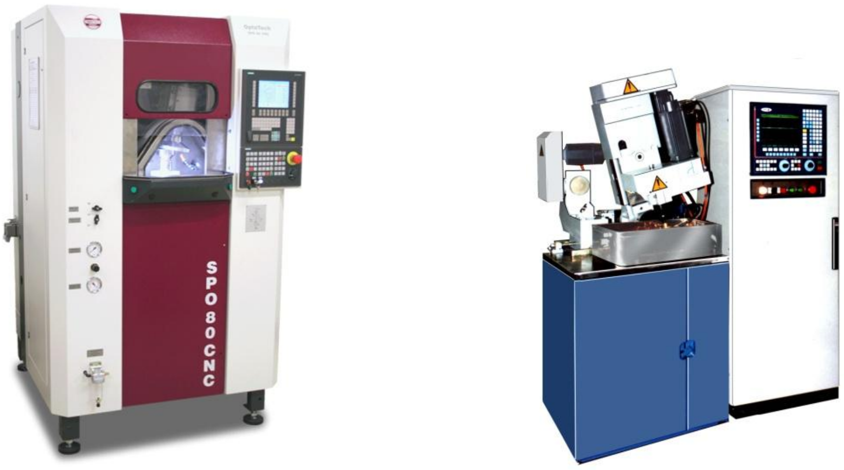


Fig. 1.3 Hybrid polishing machine "SPO 80 CNC" (left) and polishing machines for aspheric lenses "АСФЕРОИД 100П" (right)

Use ready-polishing machines:

Advantages:

- no time and production capacity for the design and construct the machine,
- documentation from the manufacturer,
- manufacturer's warranty,
- assistance service from the manufacturer,
- choice from wide range of machines suitable for the concrete tasks.

Disadvantages:

- higher costs,
- service duties,
- there is no possibility of independent upgrading (black box systems),
- there is no possibility of self-service (loss of warranty),
- polishing machines contains of excessive functions for concrete tasks.

Developing a single purpose polishing machine for a specific task from different modules can avoid these disadvantages. These machines have been designed independently from the various elements, thus have a flexible setting for specific targets. And also easily upgraded or parts of these machines can be used in the future for new projects.

Using modular polishing machines:

Advantages:

- created for a specific task (no redundancies),
- relatively low cost,
- the ability to upgrade,
- the ability to change modules,
- saving warranty for the individual modules,
- possibility of mixing modules from different manufacturers to find the optimum combination.

Disadvantages:

- Requires time, human and production capacity to design and manufacture a polishing machine.

1.2 PURPOSE (AIM)

The main aim of work, improve the existing polishing system by means of implantation modern control system inside and changing in polishing machine structural part.

Current polishing machine includes consists of a three rotation axes. The main axis is responsible realize for the rotation of the polished object (in this case an aspherical lens on polishing table). Two other axes (eccentric axis) are responsible for moving the polishing element on the lens during its rotation. These axes are driven from the main axis, which connects with them via belt or chain drive. Eccentrically axis mechanically interconnected via arms that have configurable length. Also, the eccentricity of the axes can be configured within the specified range (Fig. 1.3) [4].

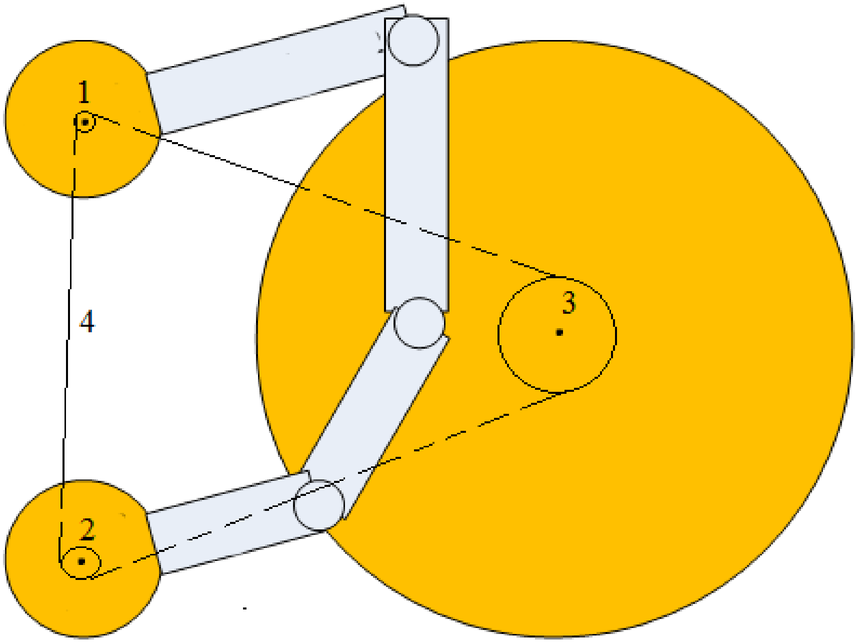


Fig. 1.3 Kinematic scheme. 1, 2 – Axis of eccentric's rotation (cams). 3 – Main axis of rotation (polishing table). 4 – Belt drive [4].

Typically, such kinds of machines are driven by one drive (usually an asynchronous motor). Main axis rotated directly from the drive, and the other two axes are connected to the main axis with belt drive. In this way, changing the rotational speed, of eccentric made manually with stopping the whole system and selecting the gears or rollers with other gear ratio or changing the eccentricity itself. Another possible way to change the speed – variator, which connects the main axis with the rest of the mechanism [5].

These methods changing of speed, is obviously, a long process that requires human intervention and stopping the polishing process for a while. Every time requires setting up the polishing machines and mechanical intervention in order to change the gear ratio or setting the precise eccentricity is time demanding.

The first step in the modernization of the system is to use independent drive for both eccentric axes. Using inverter will allow for infinite adjustment of speed for each drive in the range from zero to rated speed. Using independent drive for each axis and master control by motion controller is bringing the possibility to create electronic coupling between main axis and both eccentric axes. That brings the possibility to create an arbitrary polishing trajectory and receive the nonlinear relationship between all axes [6] [7] [8].

The second step in the modernization of the system is to implement control system, which will contain in its structure motion controller. Its gives us possibility directly control polishing head from HMI. Operator must only choose coordinates on polishing object and system will be calculated all necessary parameters and position polishing head.

Stages of work:

- Create model of system;

- Obtain direct and inverse relationship between angles of eccentric rotation and coordinates of desire point on polishing object;
- Creating a program for simulation of system work and checking system model.
- Creation of a modular polishing machine system with independent drives for each axis;
- Configuration of system;
- Development of software;

1.3 DESCRIPTION OF THE OBJECT

For describing of the object shall find dependence of the position polishing element from angle of eccentric's rotation. For this we use the placed below kinematic scheme (Fig. 1.4) [8].

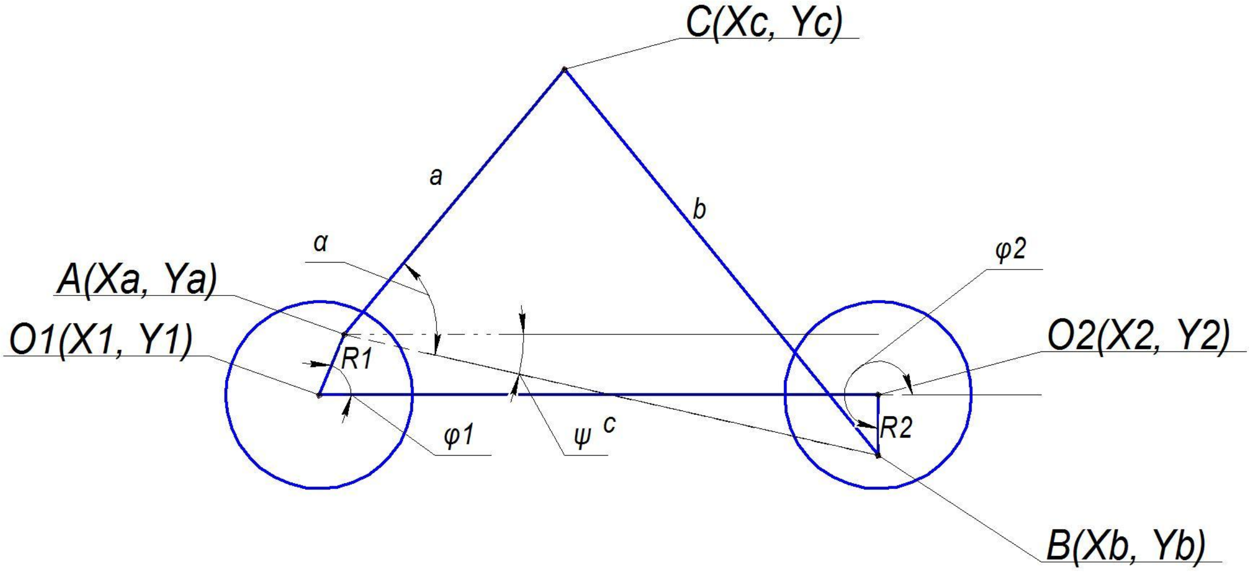


Fig. 1.4 Kinematic scheme which using for finding dependence of the position polishing element from angle of rotation eccentrics (first case)

Initially are known the following parameters (defined in the system design):

- φ_1– the angle of the first eccentric rotation,
- φ_2– the angle of the second eccentric rotation,
- O1(X_1, Y_1) – the axis of the first eccentric rotation,
- O2 (X_2, Y_2) – the axis of the second eccentric rotation,
- R_1– the length of the eccentricity of the first eccentric,
- R_2 – the length of the eccentricity of the second eccentric
- a - the length of the first arm,
- b - the length of the second arm.

Find the coordinates of the beginning of arm, depending on the rotation of eccentrics, by the following formulas:

$$\begin{cases} X_a = X_1 + \cos\varphi_1 \cdot R_1 \\ Y_a = Y_1 + \sin\varphi_1 \cdot R_1 \end{cases} \tag{1.1}$$

$$\begin{cases} X_b = X_2 + \cos\varphi_2 \cdot R_2 \\ Y_b = Y_2 + \sin\varphi_2 \cdot R_2 \end{cases} \tag{1.2}$$

Find the distance between ends of eccentricities:

$$\overline{|\mathrm{C}|} = \sqrt{(X_a - X_b)^2 + (Y_a - Y_b)^2} \tag{1.1}$$

Using the cosine theorem we find the angle α:

$$\alpha = \arccos\left(\frac{a^2 + c^2 - b^2}{2 \cdot a \cdot c}\right) \tag{1.2}$$

Amount of deviation ψ find by next formula:

$$\psi = \frac{|X_a - X_b|}{\overline{|C|}} \tag{1.3}$$

From the following equations we obtain that coordinates of the point C (X_c, Y_c), for case is shown in Fig. 1.4, can be found using the formula:

$$\begin{cases} X_c = X_a + \cos(\alpha - \psi) \cdot a \\ Y_c = Y_a + \sin(\alpha - \psi) \cdot a \end{cases} \tag{1.4}$$

This formulas will work only for case, when $X_a < X_b$ and $Y_b < Y_a$ ($X_b < X_a$ and $Y_a < Y_b$). For the case is shown in Fig. 1.5, must be used the formula (1.5).

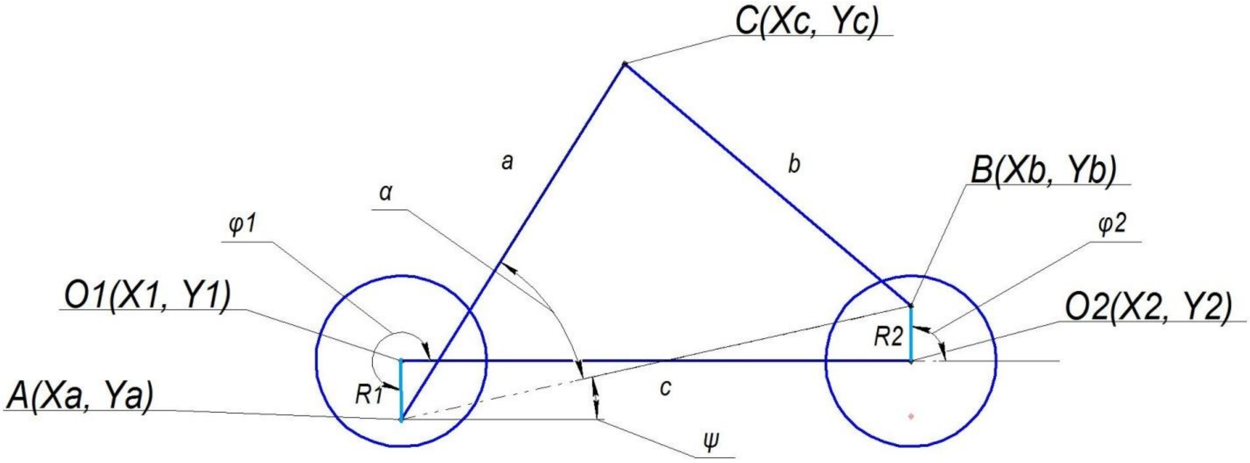


Fig. 1.5 Kinematic scheme which using for finding dependence of the position polishing element from angle of rotation eccentrics (second case)

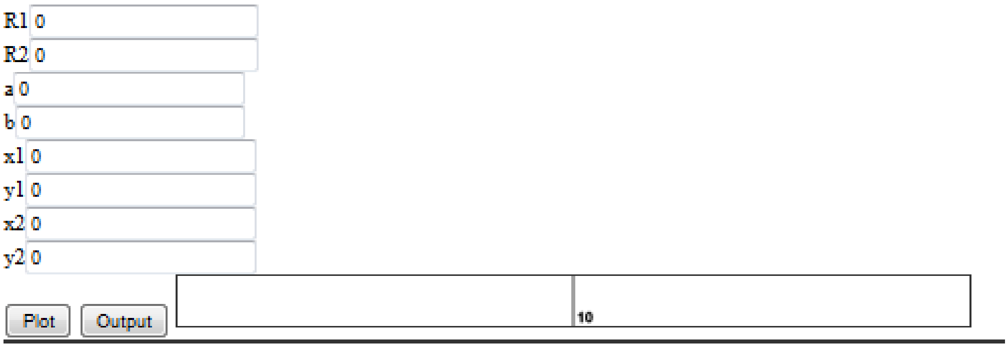
$$\begin{cases} X_c = X_a + \cos(\alpha + \psi) \cdot a \\ Y_c = Y_a + \sin(\alpha + \psi) \cdot a \end{cases} \tag{1.5}$$

Using the programming language JavaScript and formulas (1.4) and (1.5), implement a graphical representation of full possible range of polishing element position, depending on the angle of eccentrics rotation and the system configuration (parameters of moving elements) (Appendix 1). To work with the two possible states of the system will perform the following checks:

If (xb > xa && yb < ya) psi= -1;*

*Else if (xb < xa && yb > ya) psi *= -1;*

The interface is created by the program allows you to set the initial parameters of the system, graphical displayed results using the button "Plot" and get a table with calculated values using the button "Output" (Fig. 1.6)



R1 0
R2 0
a 0
b 0
x1 0
y1 0
x2 0
y2 0

Fig. 1.6 Program interface

Simulate the system with the following parameters:

- R_1 = 2
- R_2 = 0
- a = 8
- b = 6
- $O_1\ (-4, 0)$
- $O_2\ (4,0)$

In system, with such configuration, movement of the second arm will be limited by rotation around a first eccentric axis (because eccentricity is zero). The simulation results are shown in Fig. 1.7.

Graph confirms the expected results and confirmed by kinematic structure of the system. The created program have opportunity to draw the simulation result with the displaying elements a or b (or both in same time). The result for the following parameters is shown in Fig. 1.8.

- R_1 = 2
- R_2 = 0
- a = 8
- b = 6
- $O_1\ (-5, 0)$
- O_2 (5,0)

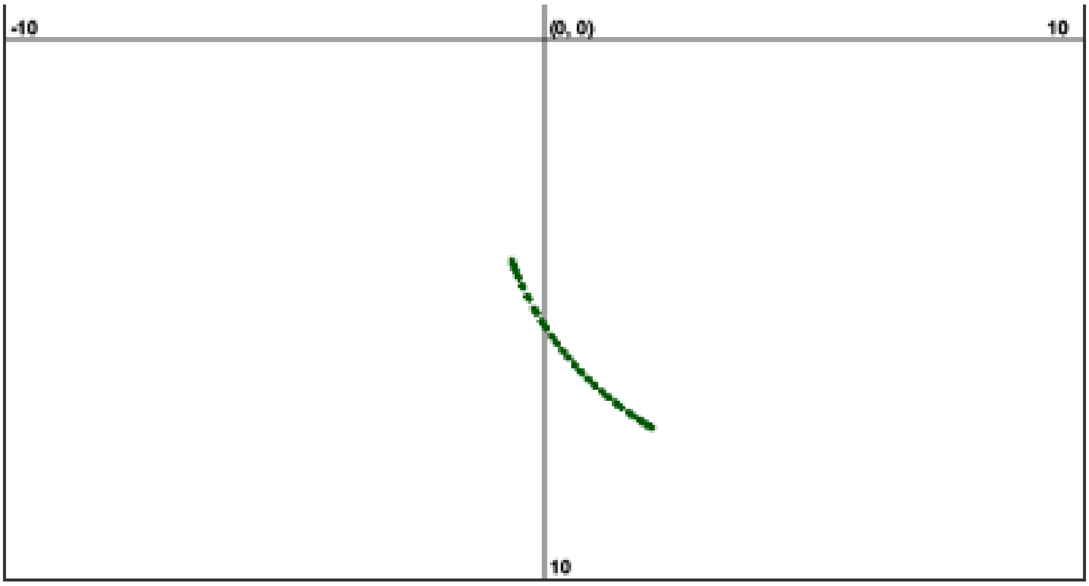


Fig. 1.7 Simulation results

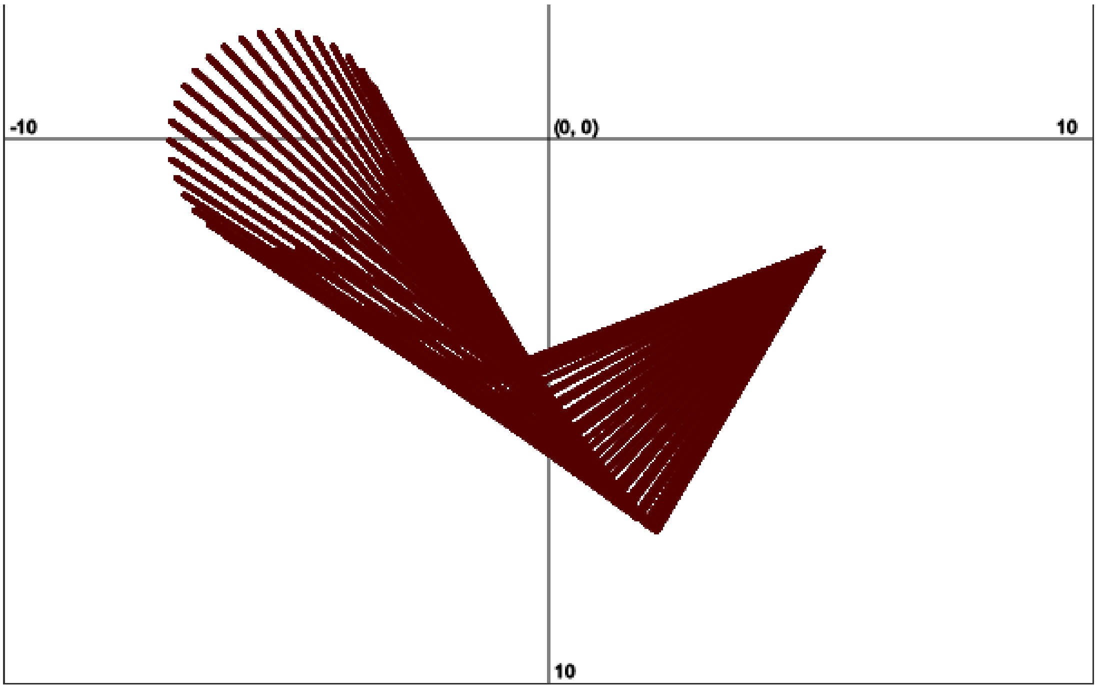


Fig. 1.8 Simulation results in the display mode with the elements a and b

Simulate the system with the following parameters:

- $R_1 = 2$
- $R_2 = 2$

- a = 8
- b = 8
- $O_1\ (-3, 0)$
- $O_2\ (3,0)$

These parameters are symmetrical and the area, resulting from simulations, must be symmetric with the axis of abscissa. The result of the simulation is shown in Fig. 1.9 and Fig. 1.10.

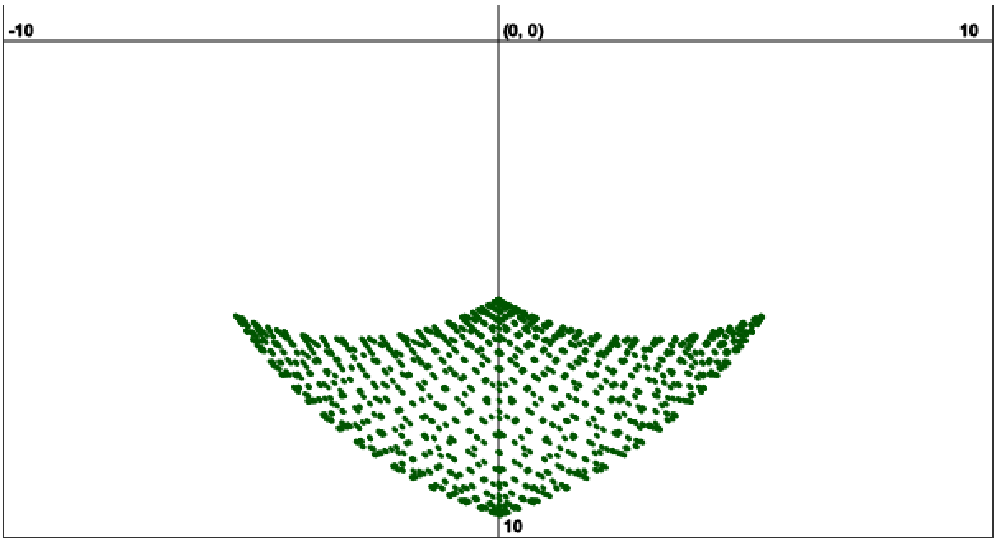


Fig. 1.9 Simulation result

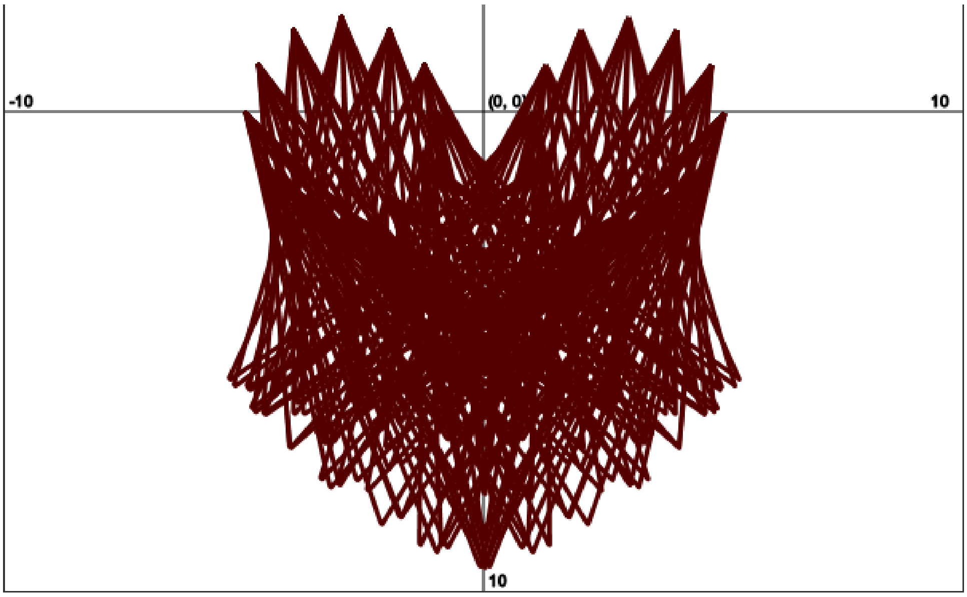


Fig. 1.10 Simulation results in the display mode of the elements a and b

Graph confirms the expected results and confirmed by kinematic structure of the system.

For positioning the polishing element needed to get the dependence equations for computing the angle of eccentric’s axes in order to set the polishing element into the concrete coordinates in two dimensional Cartesian coordinates. In this case, the description will use scheme, presented in Fig. 1.11.

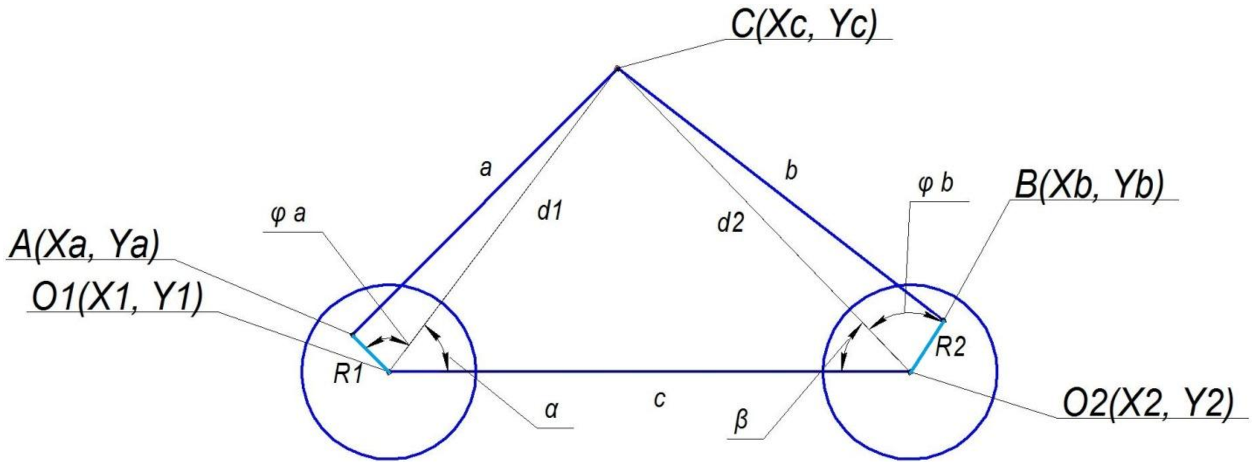


Fig. 1.11 Scheme of system

- O1(X_1, Y_1) – the axis of rotation of the first eccentric,
- O2 (X_2, Y_2) – the axis of rotation of the second eccentric,
- R_1– the length of the eccentricity of the first eccentric,
- R_2 – the length of the eccentricity of the second eccentric
- a – the length of the first arm,
- b – the length of the second arm,
- C (X_c, Y_c) – coordinates of polishing element.

To find the distance between the axes of eccentric’s rotation and the polishing element (d_1 and d_2) will use the following formula:

$$\overline{|d_1\,|} = \sqrt{(X_1 - X_c)^2 + (Y_1 - Y_c)^2} \tag{1.6}$$

$$\overline{|d_2|} = \sqrt{(X_2 - X_c)^2 + (Y_2 - Y_c)^2} \tag{1.7}$$

The distance between the axes of eccentrics:

$$\overline{|\mathrm{C}|} = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \tag{1.8}$$

Using the cosine theorem, find the angles α, β, φ_a, φ_b:

$$\alpha = \arccos\left(\frac{{d_1}^2 + c^2 - {d_2}^2}{2 \cdot d_1 \cdot c}\right) \tag{1.9}$$

$$\beta = \arccos\left(\frac{{d_2}^2 + c^2 - {d_1}^2}{2 \cdot d_2 \cdot c}\right) \tag{1.10}$$

$$\varphi_a = \arccos\left(\frac{{R_1}^2 + {d_1}^2 - a^2}{2 \cdot R_1 \cdot d_1}\right) \tag{1.11}$$

$$\varphi_b = \arccos\left(\frac{{R_2}^2 + {d_2}^2 - a^2}{2 \cdot R_2 \cdot d_2}\right) \tag{1.12}$$

The resulting dependence of angles of eccentric’s rotation from the coordinates of polishing element will be as follows for the case shown in Fig. 1.11:

$$\begin{cases} \varphi_1 = \alpha + \varphi_a \\ \varphi_2 = \pi - \beta - \varphi_b \end{cases} \tag{1.13}$$

For the case shown in Fig. 1.12, the relationship will be as follows:

$$\begin{cases} \varphi_1 = \alpha - \varphi_a + 2 \cdot \pi \\ \varphi_2 = \pi - \beta + \varphi_b \end{cases} \tag{1.14}$$

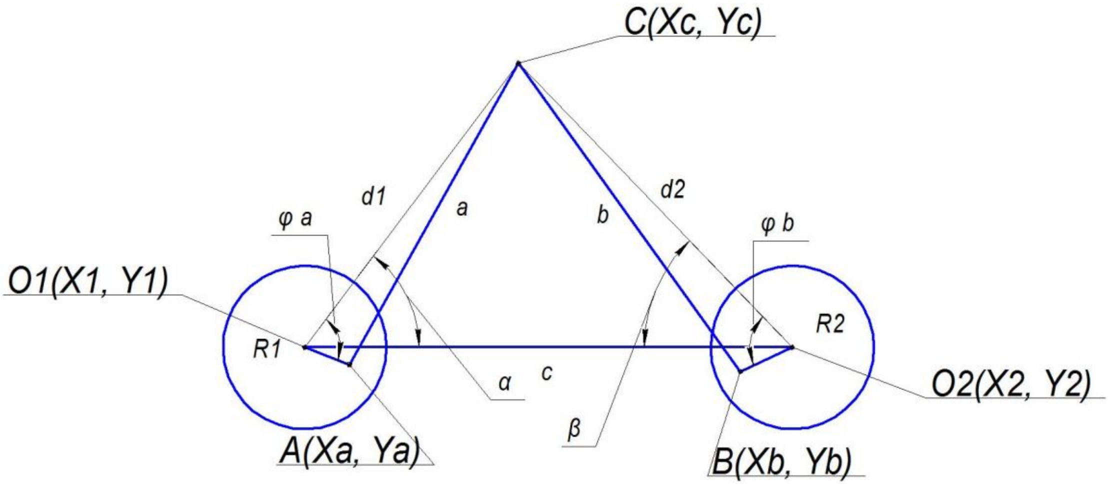
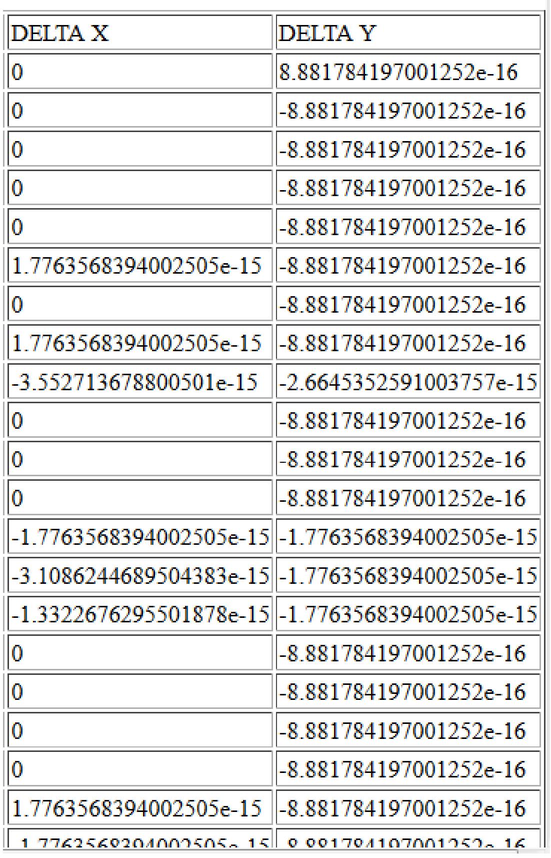


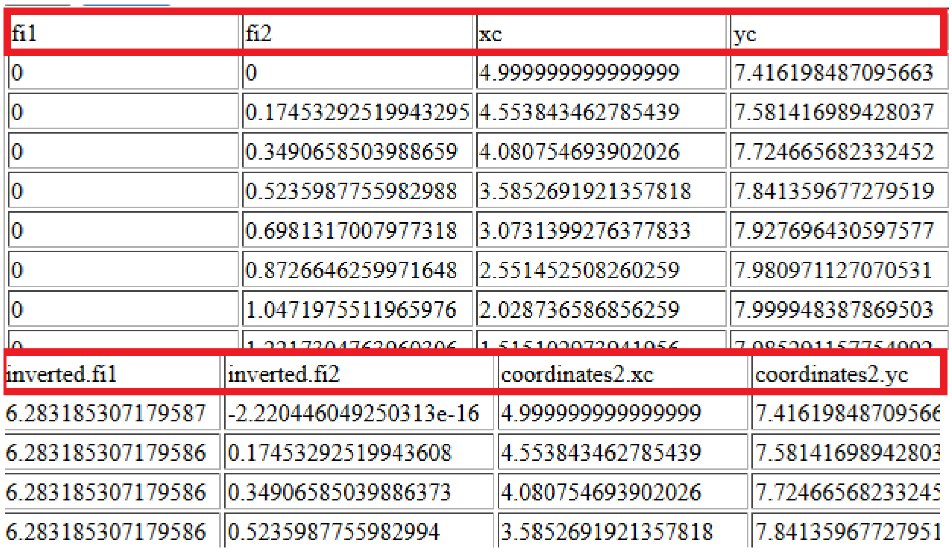
Fig. 1.12 Scheme of system

For checking, substitute the coordinates, obtained by the formula (1.5), in (1.14) and then evaluate the difference between the results (Appendix 1) Fig. 1.13a and 1.13b:



DELTA X	DELTA Y
0	8.881784197001252e-16
0	-8.881784197001252e-16
0	-8.881784197001252e-16
0	-8.881784197001252e-16
0	-8.881784197001252e-16
1.7763568394002505e-15	-8.881784197001252e-16
0	-8.881784197001252e-16
1.7763568394002505e-15	-8.881784197001252e-16
-3.552713678800501e-15	-2.6645352591003757e-15
0	-8.881784197001252e-16
0	-8.881784197001252e-16
0	-8.881784197001252e-16
-1.7763568394002505e-15	-1.7763568394002505e-15
-3.1086244689504383e-15	-1.7763568394002505e-15
-1.3322676295501878e-15	-1.7763568394002505e-15
0	-8.881784197001252e-16
0	-8.881784197001252e-16
0	-8.881784197001252e-16
0	-8.881784197001252e-16
1.7763568394002505e-15	-8.881784197001252e-16
1.7763568394002505e-15	8.881784197001252e-16

Fig.1.13a Comparison of the results



fi1	fi2	xc	yc
0	0	4.999999999999999	7.416198487095663
0	0.17453292519943295	4.553843462785439	7.581416989428037
0	0.3490658503988659	4.080754693902026	7.724665682332452
0	0.5235987755982988	3.5852691921357818	7.841359677279519
0	0.6981317007977318	3.0731399276377833	7.927696430597577
0	0.8726646259971648	2.551452508260259	7.980971127070531
0	1.0471975511965976	2.028736586856259	7.999948387869503
0	1.2217304763960306	1.515102073041956	7.985201157754093

inverted.fi1	inverted.fi2	coordinates2.xc	coordinates2.yc
6.283185307179587	-2.220446049250313e-16	4.999999999999999	7.41619848709566
6.283185307179586	0.17453292519943608	4.553843462785439	7.5814169894280
6.283185307179586	0.3490658503988 6373	4.080754693902026	7.7246656823324
6.283185307179586	0.5235987755982994	3.5852691921357818	7.84135967727951

Fig.1.13b Comparison of the results

- fi1, fi2 – angles of eccentric’s rotation. Substitute in direct formula (1.5 – 1.6).
- xc,yc – coordinates obtained by direct formula
- inverted.fi1, inverted.fi2 – angles of eccentrics rotation, which obtained by substitution xc and yc in to revise formula (1.13 – 1.14)
- coordinates2.xc, coordinates2.yc – coordinates which obtained by substitution inverted.fi1, inverted.fi2 in to direct formula.
- DELTA X, DELTA Y – difference between coordinates2.xc, coordinates2.yc and xc, yc, respectively.

Error is zero or small non-significant value associated with the feature of operations in JavaScript [9] [10].

These dependences are working for a system with a static table. We consider the case with three axes of rotation. The third rotation axis is the axis of polishing table rotation on which the object is located. Consider the scheme shown in the Fig. 1.14, to obtain the correlation between the coordinate of desired point and angle of table rotation.

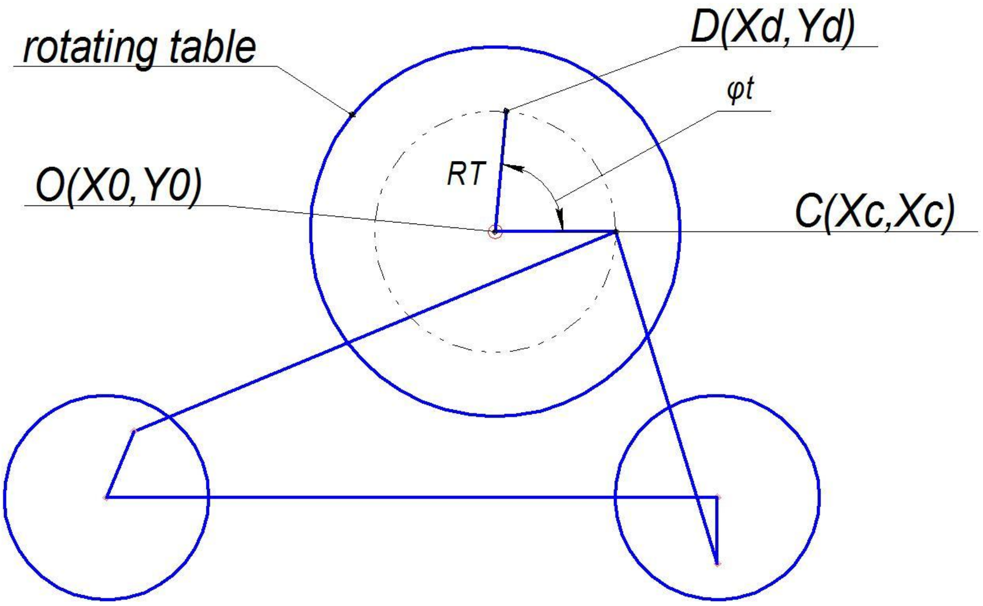


Fig. 1.14 Scheme for system with three axes

- D (X_d, Y_d) – the actual coordinates of the desired position,
- O (X_0, Y_0) – the axis of table rotation,
- C (X_c, Y_c) – the initial position of the desired point (the angle of table rotation is zero)
- φ_t – the angle of table rotation,
- R_t – distance between the axis of table rotation and desired point. Calculated using the formula (1.15) :

$$\overline{|R_t\,|} = \sqrt{(X_0 - X_c)^2 + (Y_0 - Y_c)^2} \tag{1.15}$$

The two dimensional orthogonal coordinates of polishing tool can be calculated from the actual positions of both axes can be means of following derived formulas (1.16):

$$\begin{cases} X_d = X_0 + \cos\varphi_t \cdot R_t \\ Y_d = Y_0 + \sin\varphi_t \cdot R_t \end{cases} \tag{1.16}$$

2 STRUCTURE OF APCS

To control the three axes polishing machine, by way, which was designated in subchapter 1.2, and implement, obtained in subchapter 1.3, formulas, we must to create APCS, which will contain parts described below. Our system will be separated in two levels (high and low). This is a common way of separating APCS [22].

2.1 LEVEL STRUCTURE OF APCS

The structure of created system will be as follows (Fig. 2.1). In our case PC will be used for creating software and programming PLC (motion controller). OP will be used, in the future, for controlling polishing process and like information panel for operator. On the panel will displayed information about state of the polishing process, selected operating mode and errors. In our case PLC role will be perform by motion controller. EM will be presented by three drives, which are responsible for the three axes of the polishing machines. Levels will have different kinds of communication with each other; it will be described in greater detail in subchapters 2.2 and 2.3.

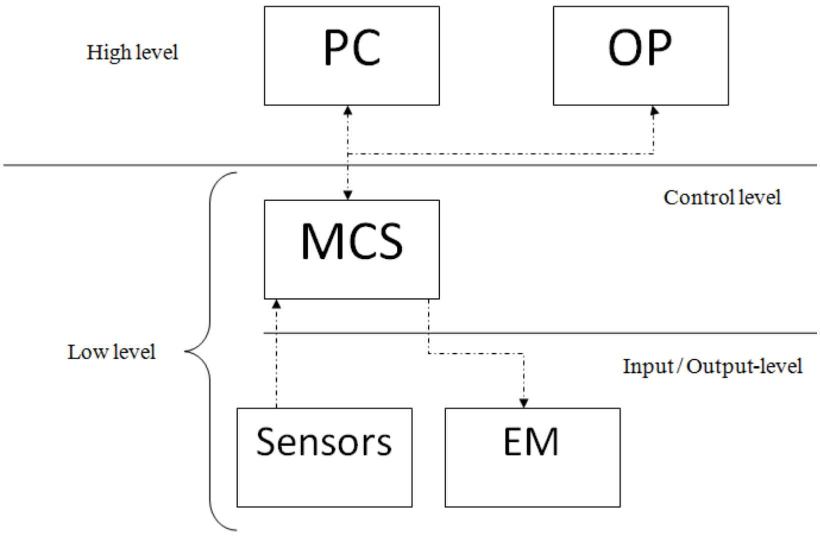


Fig. 2.1 Scheme of the structures APCS

- PC – personal computer;
- OP – operator panel;
- MCS – motion control system;
- EM – executive mechanisms.

PC – part of high level system with the installed special application software, performs the functions of display and input of information and for program the controller.

OP – takes the role of additional display device (indicator lamps and / or mnemonic scheme) and data input with a touch screen or buttons on the panel.

MSC – specialized device (devices) performs functions for process automation.

EM – Motors. Controlled by PLC according programmed algorithm in to system.

Low level – level of equipment (Input / Output-level). Level includes sensors, measuring devices, actuators, executive devices. At this level signal from the sensors communicate with the inputs of the control device and outputs signals communicate with actuators.

Control level – the PLC level. PLC receives information from the control and measuring equipment and sensors about process conditions and creates outputs control commands according with a programmed algorithm.

High level – HMI (Human Machine Interface) and SCADA (supervisory control and data acquisition) level.

2.2 COMMUNICATION BETWEEN LOW LEVEL DEVICES

Exchange of information between the low-level devices, control and measuring equipment and actuators is implemented using PROFIBUS (Process Field Bus) [11]. This is an open industrial network, which was originally developed by Siemens AG for their purposes (PLC Simatic). But in the future the network has become a common open standard.

The network makes possible to connect the various automation tools such as sensors, actuators, logic devices into a single system. By means PROFIBUS automation elements from different manufacturers can communicate with each other without additional special interfaces. Network interface represents chip ASIC (application-specific integrated circuit), which produce a lot of manufacturers.

PROFIBUS network based on RS 485 interfaces. The network constructed on the basis of this interface represented by device receiving and transmitting signals, interconnected by twisted pair. In this case, the data exchange is implemented between one master and several slaves.

Characteristics of the RS 485 [12]:

- The number of transmitters - 32;
- Number of receivers - 32;
- Maximum communication speed - 10 Mbit / s (12Mbit / s using additional units);
- The voltage range of the logical unit of the transmitter from +1.5 to +6 V;
- The voltage range of the transmitter logic zero from -6 to + 1.5 V;
- Maximum cable length 1,2km;
- The maximum load impedance of the transmitter 54 Ohm;

Advantages of the RS 485 interface [12]:

- Ability to create bus or tree topology using repeaters, bus terminals and bus connectors for connecting to the network devices with PROFIBUS interface.
- Passive signaling allows disabling nodes without affecting to the state of the network.
- Relatively simple installation and cable connection.

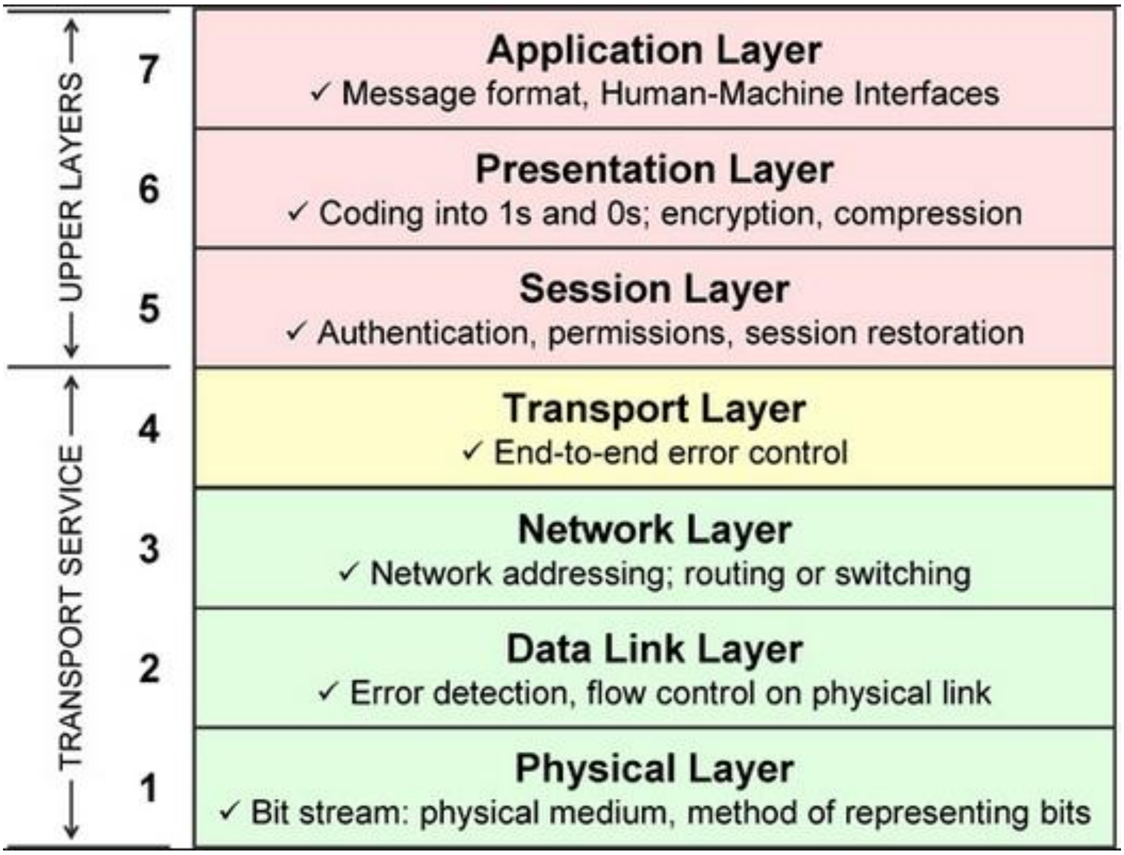
Limiting the RS-485 interface:

- Increasing the length of the network creating decreasing data transfer rate.

Profibus DP (Decentralized Peripherals) - Profile protocols industrial network Profibus.

The used layers of OSI (Open Systems Interconnection) (Fig. 2.1) model [13]:

- Layer 1: physical layer (Transmission and reception of raw bit streams over a physical medium).
- Layer 2: data link layer (Reliable transmission of data frames between two nodes connected by a physical layer)
- Layer 7: application layer (High-level APIs, including resource sharing, remote file access, directory services and virtual terminals)



UPPER LAYERS	7	**Application Layer** ✓ Message format, Human-Machine Interfaces
	6	**Presentation Layer** ✓ Coding into 1s and 0s; encryption, compression
	5	**Session Layer** ✓ Authentication, permissions, session restoration
TRANSPORT SERVICE	4	**Transport Layer** ✓ End-to-end error control
	3	**Network Layer** ✓ Network addressing; routing or switching
	2	**Data Link Layer** ✓ Error detection, flow control on physical link
	1	**Physical Layer** ✓ Bit stream: physical medium, method of representing bits

Fig. 2.2 OSI layers [13]

2.3 COMMUNICATION BETWEEN HIGH AND LOW LEVEL DEVICES

Communication between the components of the low and high levels carried out mainly by means of industrial computer network – Industrial Ethernet, provides a full cycle of data exchange between components within one second [14].

Industrial Ethernet a variant of the standardized Ethernet, used in industrial networks for connecting in low and high levels. Much rarer Industrial Ethernet used to exchange information between the PLCs and their subordinate devices (sensors, actuators, motors). The reason why the Industrial Ethernet rarely used for the last purpose, in essence it is used to access procedures SSMA / CD, not guaranteeing a small number of bytes exchanged with sufficiently high frequency [15].

Distinctive features of Industrial Ethernet:

- Cables and Wiring connectors have standard, suitable for industrial networks (increased service life, high shielding) ;
- Availability of suitable standards for communication with mobile objects (flexible cable);
- Changed the protocol stack TCP / IP allows the transfer of small amounts of data with high frequency (this type of data exchange is typical for programmable logic controllers);
- With additional switches in the network can be implemented Rapid Spanning Tree Protocol which, unlike conventional Spanning Tree Protocol, finds reserve communication path much faster (additional fault tolerance) [14].

3 SOFTWARE SELECTION

Software Requirements:

- Simple integration with selected hardware;
- Ability to configure the system;
- Creation of a communication network and its topology;
- Motion Control;
- Creation of different trajectories of movements during simultaneous operation of several drives;

To configure the motion controller can be uses a special program – SIMOTION SCOUT [15]. SCOUT engineering software it is a powerful tool for Simotion controller programming, which has integration for all technology products in the concept of Siemens – TIA (Totally Integrated Automation). SCOUT software allows to work with the project from single software product and contains tool for commissioning drive that allows you to enter configuration data, programs, motion profiles, settings and so on.

SCOUT also has an extensive system testing, debugging and diagnostics programs, which includes displaying the status of the program, displaying the current values of all variables, control variables when executing commands in real-time, electronic oscilloscope, control panel and drive axles, etc [16]. SCOUT project navigator contains all the necessary development tools and it’s a part of interface from which possible performs all stages of the project. SCOUT software package is installed as an addition pack to the Simatic Manager and uses libraries from the main program. Separately using SCOUT is not possible.

Advantages of using SIMOTION SCOUT:

- User friendly and intuitive interface through integrated functionally oriented tools;

- Centralized data and project management (even in distributed systems);
- Quick access to individual configuration tools, programming, parameter setting, etc;
- Orientation to the automation of the process associated with the motion.

For installing SIMOTION SCOUT, on PC, previously, must be installed basic program Simatic Step 7 Manager. Especially for families of controllers S7-300 / 400 (and based on them), Siemens has developed its own software Simatic Step 7 which includes many different tools. In our case will be used motion controller Simotion C, which based on S7-300. The program allows carrying out complex work on the design and maintenance of automation systems, based on programmable logic controllers Simatic S7-300 and Simatic S7-400. The main work on the project takes place in the utility Simatic Manager [15] [16] [17].

During system configuration, via SIMOTION SCOUT, defined used modules, methods of communication, topology and configuration of modules. Software can checks connected to the network components and their settings. The final step its load the settings of the selected configuration in the hardware.

Additional tools allow [17]:

- Identify hardware errors;
- Perform the diagnostics of the equipment;
- Identify the incorrect installation of the equipment.

Programming of controller takes place in editor of program, which provides the ability to write programs in the five main languages. In addition, the editor allows you to monitor the current state of the program and parameters, regardless of the programming language, thus providing the ability to debug software.

To creating an arbitrary polishing trajectory and receive the nonlinear relationship between all axes can be used Simotion Scout package “Path Interpolation” and “Cam” [23] or, like in our case, we will prepare all formulas using structure language and it let avoid limitations of system package (max 3 axis and limited choice of system structure).

4 DEVELOPMENT

4.1 SYSTEM CONFIGURATION

In this test system, the rotation of eccentrics is performed with a real drive (selected in "Drive"), and the rotation of the polishing table is implemented using a virtual axis. In the future the system will be extended with HMI via the operator panel. The general structure of the final system is shown in Fig. 4.1.

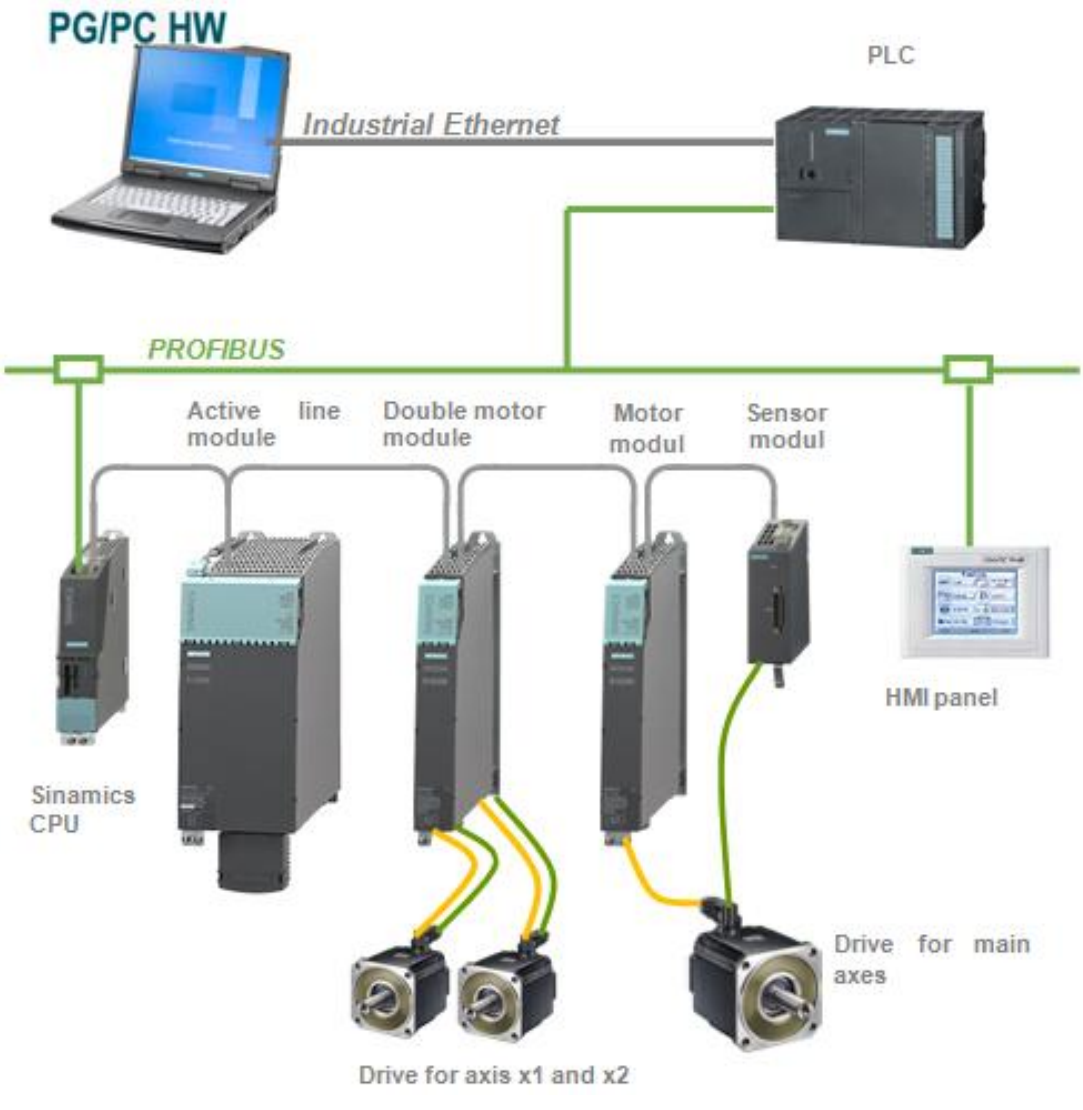


Fig. 4.1 General structure of the final system

To create a motion control system based on hardware and software package SIMOTION automation, you need the following components:

- PC,
- PLC (Motion controller),
- Control Unit;
- Line Module;
- Motor Module;
- Drives.

PC

The PC is used for programming logic controller, via installed on it application software (Simatic Manager,Simotion Scout). Requirements for the characteristics a PC are shown in Table 4.1 [17].

Table 4.1 Requirements for PC's characteristics

Operating system	Windows XP x32, Windows 7 (x32+x64)
Minimum system requirements	Pentium III 500 MHz (Windows XP) 256 MB RAM
Required interfaces	Ethernet/USB

PLC

This system will be used the PLC SIMOTION C 240 PN – this controller with motion control functions, which has the design concept in the SIMATIC S7-300 versions. In version PN it has 4 PROFINET port. SIMOTION C240 can be equipped with additional periphery modules. Regardless of performance, SIMOTION C controller has two independent interface PROFIBUS, which support the profile PROFIdrive. Requirement for the PLC and its characteristics are shown in Table 4.2 [18].

Table 4.2 Requirement for the PLC's characteristics

Requirement for the PLC	SIMOTION C 240 PN
Integrated Ethernet interface	+
Integrated PROFIBUS DP interface	+
Number of controlled axes (drives) - 3	≤ 32
Open TCP / IP communication	+
The minimum cycle adjustment of position ~ 1 ms	0,5 ms

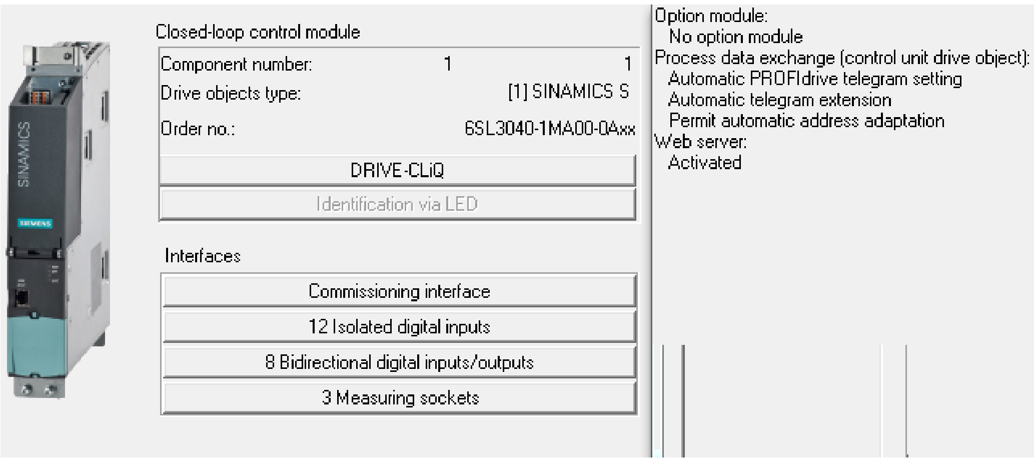
Motion Control unit

To realize motion control functions and combine different drive in one system need to use motion control unit. Its extend opportunities of controlling drives.

The created system will be used the motion control module series S120 CU320DP (Fig. 4.1), supports connectivity to PLC via PROFIBUS DP and allows to control up to 6 drives [18].The outstanding system features of the SINAMICS S120 provide the following advantages and meet the requirements of our system:

- A standard PROFIBUS or PROFINET interface and various analog and digital interfaces enable easy integration into automation solutions.
- Individual modules and power components can be replaced quickly and easily, which ensures a higher level of plant availability.
- The inverters on the motor side are coupled through a common DC link and allow energy to be exchanged between motors that are motoring and generating. In this way, energy is saved, the line infeed is relieved and line harmonics reduced.
- Simple commissioning thanks to the menu-driven STARTER commissioning tool.

- All device modules are easily accessible, which makes them extremely service friendly.
- Space-saving design



Closed-loop control module

Component number: 1 1
Drive objects type: [1] SINAMICS S
Order no.: 6SL3040-1MA00-0Axx

Interfaces

Option module:
No option module
Process data exchange (control unit drive object):
Automatic PROFIdrive telegram setting
Automatic telegram extension
Permit automatic address adaptation
Web server:
Activated

Fig. 4.2 Module Specifications S120 - CU320DP

Line module

This system will be used the active line module (order number: 6SL3130-7TE21-6Axx), whose characteristics are given in Table 4.3. Active Line Modules (ALM) can supply energy and return regenerative energy to the supply system. In contrast to Basic Line Modules and Smart Line Modules, Active Line Modules generate a controlled DC voltage that is kept constant despite fluctuations in the line voltage (the line voltage must remain within the permissible tolerance range). Active Line Modules draw a virtually sinusoidal current from the supply system and therefore do not cause any harmful current harmonics.

Table 4.3 ALM’s characteristics

Rated power	16 kW
Rated current	27 A
Input voltage	400 V
The frequency of the input voltage	50-60 Hz
Line filter	400 V, 16 kW

Motor Module

In the series SIMOTION S presented module (order number: 6SL3120-2TE21-0A00) which can control in same time two drives. This module is used to control the drives which rotate eccentrics. Characteristics of the selected module are shown in Table 4.4.

Table 4.4 Characteristics of double motor module

Module type	Double motor module
Rated power	4,8 kW
Rated current	9 A / 9 A

Drives

To rotate the eccentrics will be used drives with the following order numbers: 1FK7063-2AF71-1BGO and 1FT7062-1AF70-1NH1. Their characteristics are shown in Tables 4.5 and 4.6. These drives are compatible with the other selected modules. One of the drive is equipped with the DRIVE-CLiQ[19] system, for second drive will be used additional sensor module.

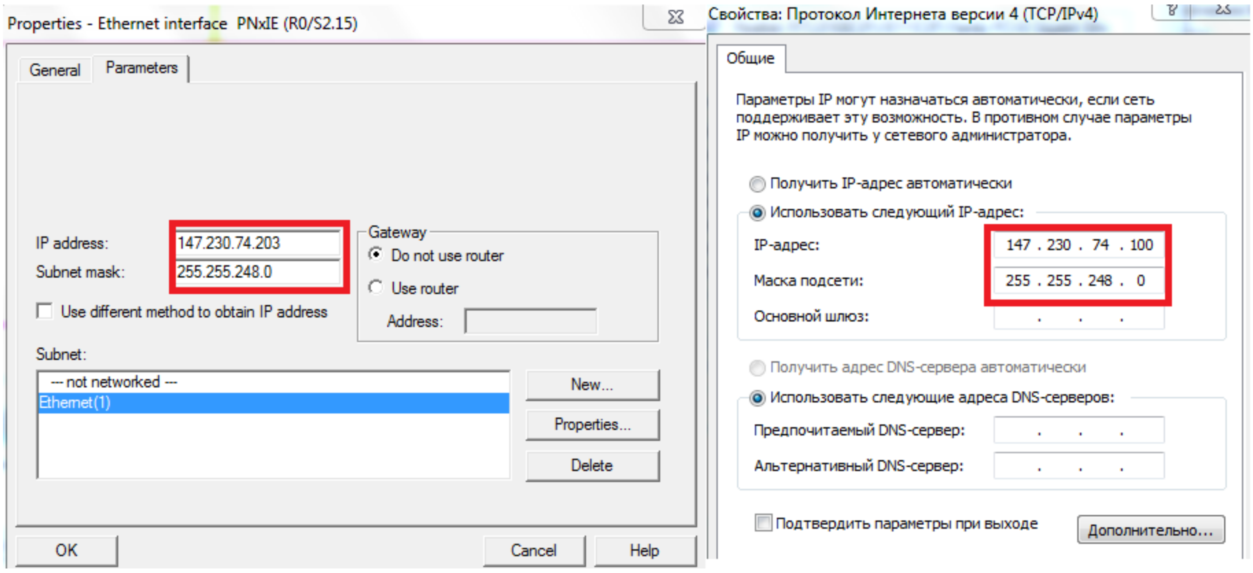
Tables 4.5 Characteristics of first drive

Type of drive	Synchronous Motor
Rated speed	3000 U/min
Maximum speed of rotation	6200 U/min
Rated torque	5,4 Nm
Rated current	3,9 A
Motor holding brake	+
DRIVE-CLiQ system	+

Tables 4.6 Characteristics of second drive

Type of drive	Synchronous Motor
Rated speed	3000 U/min
Maximum speed of rotation	7200 U/min
Rated torque	7,4 Nm
Rated current	6,6 A
Motor holding brake	+
DRIVE-CLiQ system	–

Communication between the PC and PLC occurs via Industrial Ethernet. In the setting of the parameters PC network adapter installed ip address/subnet mask which must correspond with PLC settings (Fig. 4.2). These settings are selected using the tool “Find accessible node” in Simotion Scout.



Properties - Ethernet interface PNxIE (R0/S2.15)

General | Parameters

IP address: 147.230.74.203

Subnet mask: 255.255.248.0

□ Use different method to obtain IP address

Gateway

● Do not use router

○ Use router

Address:

Subnet:

--- not networked ---

Ethernet(1)

Свойства: Протокол Интернета версии 4 (TCP/IPv4)

Общие

Параметры IP могут назначаться автоматически, если сеть поддерживает эту возможность. В противном случае параметры IP можно получить у сетевого администратора.

○ Получить IP-адрес автоматически

● Использовать следующий IP-адрес:

IP-адрес: 147 . 230 . 74 . 100

Маска подсети: 255 . 255 . 248 . 0

Основной шлюз: . . .

○ Получить адрес DNS-сервера автоматически

● Использовать следующие адреса DNS-серверов:

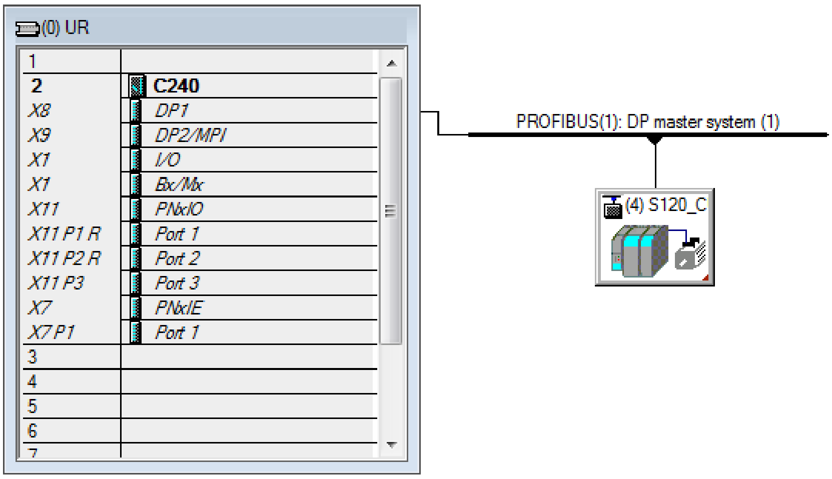
Предпочитаемый DNS-сервер: . . .

Альтернативный DNS-сервер: . . .

□ Подтвердить параметры при выходе

Fig. 4.3 Settings for Communications PLC and PC

For communication between the PLC and the control unit will be used PROFIBUS DP. For this purpose on the port DP1 on PLC is created DP master system and connected the module S120 CU320DP. Address PLC - 2, control unit address – 4 (it is set by a mechanical switch on the control unit module) Fig. 4.3.



(0) UR

1	
2	**C240**
X8	*DP1*
X9	*DP2/MPI*
X1	*I/O*
X1	*Bx/Mx*
X11	*PNxIO*
X11 P1 R	*Port 1*
X11 P2 R	*Port 2*
X11 P3	*Port 3*
X7	*PNxIE*
X7 P1	*Port 1*
3	
4	
5	
6	
7	

Fig. 4.4 Communication between the PLC and the Control unite

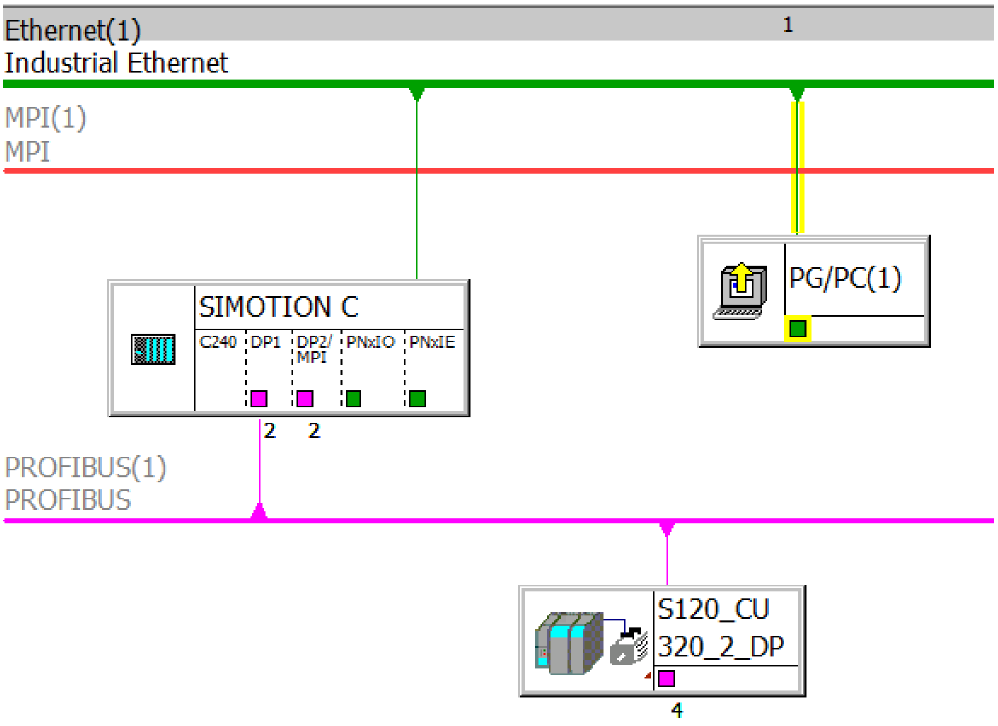
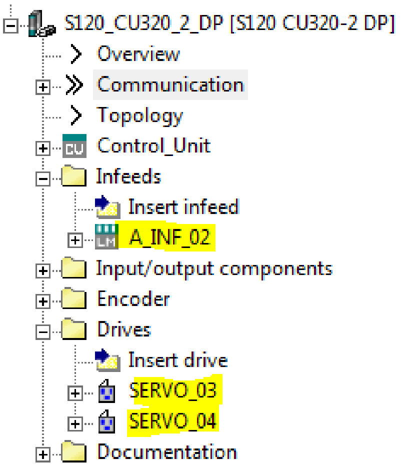


Fig. 4.5 General structure of communication in system

The active line module, the drive and control unite module equipped with DRIVE-CLiQ [19], which allows after connection of the modules configure the system and to read parameters of drives and active line module in automatic mode. The result of automatic configuration is shown in Fig. 4.5.

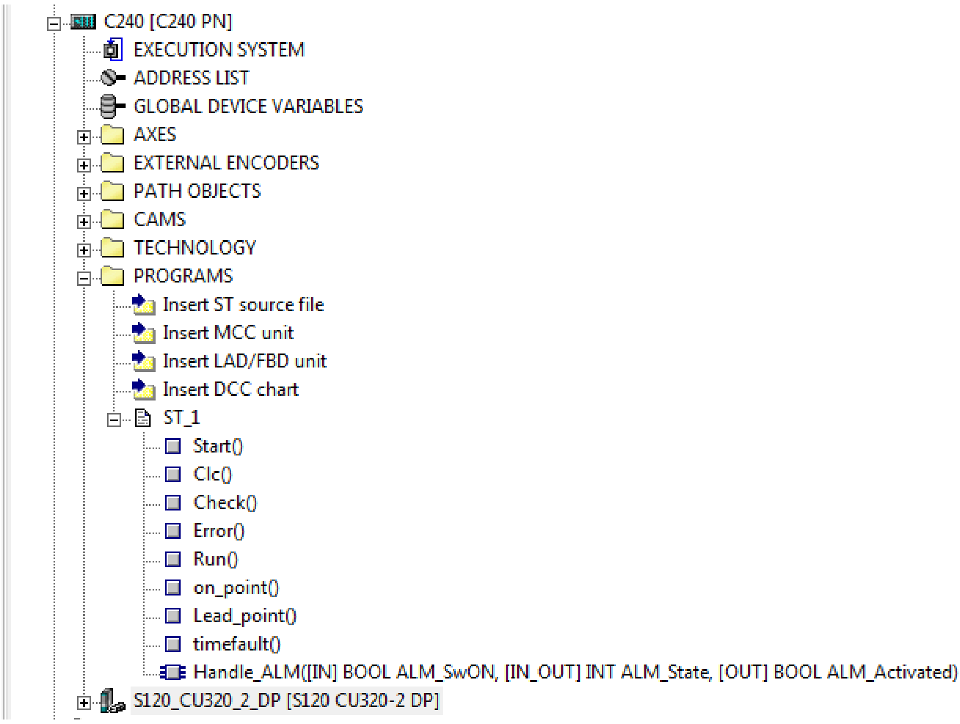


- S120_CU320_2_DP [S120 CU320-2 DP]
  - Overview
  - Communication
  - Topology
  - Control_Unit
  - Infeeds
    - Insert infeed
    - A_INF_02
  - Input/output components
  - Encoder
  - Drives
    - Insert drive
    - SERVO_03
    - SERVO_04
  - Documentation

Fig. 4.6 Result of the automatic configuration

4.2 DEVELOPMENTS OF FUNCTIONAL BLOCKS AND FUNCTIONS

For creating main program will be used ST language [20]. For this purpose in tab “PROGRAMS” will create element ST_1. Into workspace define all future function and functional block. Result is shown in Fig. 4.1.



```
C240 [C240 PN]
    EXECUTION SYSTEM
    ADDRESS LIST
    GLOBAL DEVICE VARIABLES
    AXES
    EXTERNAL ENCODERS
    PATH OBJECTS
    CAMS
    TECHNOLOGY
    PROGRAMS
        Insert ST source file
        Insert MCC unit
        Insert LAD/FBD unit
        Insert DCC chart
        ST_1
            Start()
            Clc()
            Check()
            Error()
            Run()
            on_point()
            Lead_point()
            timefault()
            Handle_ALM([IN] BOOL ALM_SwON, [IN_OUT] INT ALM_State, [OUT] BOOL ALM_Activated)
    S120_CU320_2_DP [S120 CU320-2 DP]
```

Fig. 4.7 List of function and functional block

Task, which must be performed by main program:

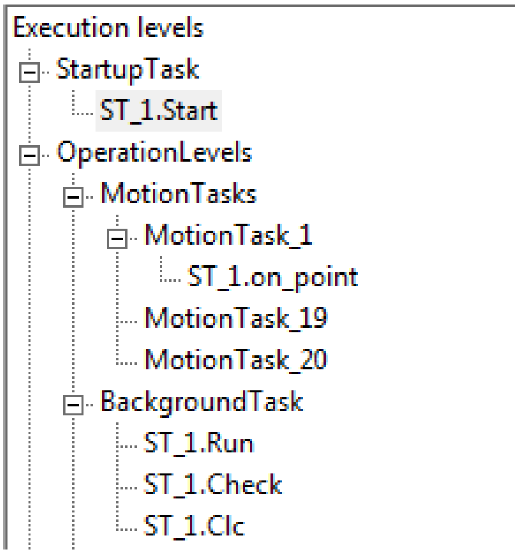
- Read input variables;
- Check input variables;
- Calculated required variables;
- Control Active line module;
- Enable drive;
- Positioning of the polishing element via rotating eccentric’s axis.

Execution system in SIMOTION SCOUT has different tasks level (Fig. 4.2):

The Startup Task is executed once at the transition from STOP or STOP U mode to RUN mode. It is intended for initialization and resetting of technology objects. In this level will be executed function “*Start*”.

The Background Task is provided for the programming of cyclic sequences without a fixed time frame. The Background Task is started after system start-up and then executed cyclically and free-running. In this level will be executed functions “*Run*”, “*Clc*” and “*Check*”. “*Check*” function nonstop performed checking of input variables. “*Clc*” function calculated variables depending on input parameters. “*Run*” function wait switching of command flag and performed corresponding action (like turn on ALM or Enable drives)

Motion Tasks are intended for the programming of sequences, for programmed motion control or other sequential executions. Motion Tasks are started by user programs and executed once. On this level will call conditionally function “*on_point*”. Condition – switching corresponding flag on TRUE value.



Execution levels
- StartupTask
  - ST_1.Start
- OperationLevels
  - MotionTasks
    - MotionTask_1
      - ST_1.on_point
    - MotionTask_19
    - MotionTask_20
  - BackgroundTask
    - ST_1.Run
    - ST_1.Check
    - ST_1.Clc

Fig. 4.8 Execution system

Function “*Check*”

The design parameters of the system are entered by the operator before starting work. Parameters should be checked for correctness (correspond the geometric structure of the system and not to create unrealizable mathematical operations in further calculations). Arm length and eccentricity, as well as the distance between the axes of eccentrics rotation, must be such that the arms would have a common point (at this point it will be located polishing element). From the formulas 1.9 - 1.12 were obtained conditions for our parameters, allowing in advance avoid division by 0 and not obtain incorrect values for computed variables. Description of input and output variables block is shown in Table 4.7.

Function working algorithm:

Step 1: Calculation of the variable d1. If the variable is not zero, move to the next step. If the variable d1 is equal to zero – end of the block work. Variable Correct_par stays in to False value.

Step 2: Calculation of the variable d2. If the variable is not zero, move to the next step. If the variable d2 is equal to zero – end of the block work. Variable Correct_par stays in to False value.

Step 3: Calculation of the variable C. If the variable is not zero, move to the next step. If the variable C is equal to zero – end of the block work. Variable Correct_par stays in to False value.

Step 4: Checking the parameter R1 for inequality to zero. If the parameter is not zero, go to the next step. If the parameter is zero – end of block work. Variable Correct_par stays in to False value.

Step 5: Checking the parameter R2 for inequality to zero. If the parameter is not zero, go to the next step. If the parameter is zero – end of block work. Variable Correct_par stays in to False value.

Step 6: Compare the variable C and the sum of arm lengths and eccentricities. If the variable C less or equal – move to the next step. If the variable C is greater – end of block work. Variable Correct_par stays in to False.

Step 7: Switch Correct_par variable to True value. End of block work.

Table 4.7 Variables used in the function "Check"

Name	Data type	Default value	Type of variable	Comment
a	Real	Sets by operator	Input	Length of arm a
b	Real	Sets by operator	Input	Length of arm b
R1	Real	Sets by operator	Input	Length of the eccentricity R1
R2	Real	Sets by operator	Input	Length of the eccentricity R2
X1	Real	Sets by operator	Input	Coordinate of the first eccentric axis
Y1	Real	Sets by operator	Input	Coordinate of the first eccentric axis
X2	Real	Sets by operator	Input	Coordinate of the second eccentric axis
Y2	Real	Sets by operator	Input	Coordinate of the second eccentric axis
Correct_par	Bool	False	Output	1 – entered parameters are correct 0 – input parameters are incorrect
Not_zero	Bool	False	Output	1 – Entered variables will lead to division by zero
Zero_error	String	0	Output	Information about error
d1,d2	Real	0,0	Local	The distance between desired position and eccentric

Function “*Clc*”

This block calculates the angles of eccentric’s rotation, which are required to achieve with polishing element desired point on the polished object. In case, if polishing table is rotated, block recalculates coordinates of the desired point in depending from the angle of polishing table rotation. For calculation used formulas 1.9 - 1.12. Description of input and output variables is shown in table 4.8.

Function working algorithm:

Step 1: Checking variable Correct_par. If value True, move to the next step. If value False, end of block work.

Step 2: Calculate the distance between the axis of the polishing table rotation and desired point on the polished object. Move to the next step.

Step 3: Calculate the new desired point coordinate in depending on angle of the polishing table rotation. Move to the next step.

Step 4: Calculation of distance between axes of the eccentric’s rotation and the desired point on the polished object. Move to the next step.

Step 5: Calculate the distance between the axes of rotation of the eccentrics. Move to the next step.

Step 6: Computation of local variables α and β. These variables are the two angles of a triangle formed by the axes of the eccentric’s rotation and the desired point. Move to the next step.

Step 7: Calculation of local variables φ_a and φ_b. When choosing a configuration shown in Fig. 1.12, move to step 8a. When choosing a configuration shown in Fig. 1.13, move to step 8b.

Step 8a: Calculate the angles of the eccentric’s rotation using the formula 1.13. End of block work.

Step 8b: Calculate the angles of the eccentric’s rotation using the formula 1.14. End of block work.

Table 4.8 Variables used in the function “Clc”

Name	Data type	Default value	Type of variable	Comment
a	Real	Sets by operator	Input	Length of arm a
b	Real	Sets by operator	Input	Length of arm b
R1	Real	Sets by operator	Input	Length of the eccentricity R1
R2	Real	Sets by operator	Input	Length of the eccentricity R2
X1, Y1	Real	Sets by operator	Input	Coordinate of the first eccentric axis
X2, Y2	Real	Sets by operator	Input	Coordinate of the second eccentric axis
Correct_par	Bool	False	Input	1 - entered parameters are correct 0 - input parameters are incorrect
d1,d2	Real	0,0	Local	The distance between desired position and axis of eccentric rotation
r_t	Real	0,0	Local	The distance between the axis of the polishing table rotation and desired point on the polished object
fi_t	Real	0,0	Input	Angle of the polishing table rotation
Xd, Yd	Real	0,0	Local	Coordinate of desire point
α, β, φ_a, φ_b	Real	0,0	Local	–

φ_1, φ_2	Real	0,0	Output	Angles of eccentrics rotation

Control Active Line Module (*Handle_ALM* block)

To control the active line module will be used built-in function block *_ALM Control*. You can use *the _ALM_control* FB [19][20] from your user program to enable and disable the Active Line Module (ALM). The *_ALM_control* FB transfers the commands to the ALM, reads the provided response data and monitors the status signals of the ALM.

A rising edge at the enable input parameter activates the Active Line Module in the error-free state. The switch from TRUE to FALSE at the enable input parameter disables the ALM. Errors present at the error output parameter must be acknowledged with a rising edge at the reset input parameter. Only the error is acknowledged. After a successful error acknowledge, the ALM must be reactivated with a rising edge at the enable input parameter. The ALM will not be enabled when errors are present (error = TRUE) and a rising edge at the enable input parameters. The error must be acknowledged first. Description of input and output parameters is shown in table 4.9 [21].

Table 4.9 Variables used in the function block "Handle_ALM"

Name	Data type	Default value	Type of variable	Comment

enable	Bool	False	Input	Enable/disable ALM
				TRUE = enable
				FALSE = disable

reset	Bool	False	Input	Acknowledge pending errors with rising edge

periIn	Word	0	Input	Input variable for the ALM status word
done	Bool	False	Output	Acknowledge enable for enable and disable. Signal present for a block call.
error	Bool	False	Output	ALM error status. TRUE = error during the request processing
errorID	Dword	0	Output	Specification of the error For error = TRUE, the errorID parameter contains the error information (refer to the "Error messages" table)

activated	Bool	False	Output	ALM status:
				TRUE = ALM enabled
				FALSE = ALM disabled

periOut	Dword	0	Output	Output variable for the ALM control word

CONCLUSION

In this work I have developed and described the testing stand, including hardware and software aspects of the design. The main goal of my master thesis was fulfilled. Obtained stand and formulas can be used in future work, which connected with innovation of the aspherical lenses polishing machine or two eccentric’s axis machine.

Written in this work script allows you to simulate operation of system with different parameters and calculated required variables, and, also, graphically present the results of the simulation. The script was written in the well known programming language JavaScript, and it makes possible easily changed script in future for a variety of task, associated with such systems.

Configured system allows you to manage the drives in accordance with the obtained formulas; the individual modules can be easily replaced, so you can use the created system with different hardware configurations.

REFERENCE

[1] R. Schwarte, O. Loffeld, F. Klaus, H.Heinol, R. Klein, — Power transmittance of optimized aspherical lens with large numerical aperture, SPIE Vol. 2775.

[2] Bernhard Braunecker, Rüdiger Hentschel, Hans J. Tiziani (Hrsg.): Advanced Optics Using Aspherical Elements SPIE Press, 2008, ISBN 978-0-8194-6749-2.

[3] Warren J. Smith, Modern optical engineering, 3rd ed., McGraw-Hill (2000).

[4] Josef Černohorský, David Lindr, Zpráva o řešení –Návrh systému stroje LOH, 2014.

[5] Spitzer, David W. (1990). Variable Speed Drives, Instrument Society of America, ISBN 1-55617-242-7.

[6] SIMOTION Technology Objects Synchronous Operation, Cam, Function Manual, Edition 2008.

[7] SIMOTION Motion Control TO Path Interpolation, Function Manual, Edition 2014.

[8] Craig J.,J.: Introduction to Robotics Mechanics and Control, Third Edition, Prentice Hall, 2005.

[9] William Kahan, Lecture Notes on the Status of IEEE Standard 754 for Binary Floating-Point Arithmetic, 1987. Available:
http://www.cs.berkeley.edu/~wkahan/ieee754status/IEEE754.PDF, [Online]. [Accessed 1 Apr 2015].

[10] David Goldberg, What Every Computer Scientist Should Know About Floating-Point Arithmetic, 1991, [Online]. Available:

http://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html.[Accessed 5 Apr 2015].

[11] J. Weigmann, G. Kilian, Decentralization with PROFIBUS DP, ISBN 978-3-89578-218-3.

[12] RS-485, [Online]. Available: https://ru.wikipedia.org/wiki/RS-485 [Accessed 9 May 2015].

[13] The OSI Model's Seven Layers Defined and Functions Explained, Microsoft Support, 2014, [Online]. Available: https://support.microsoft.com/ru-ru/kb/103884 [Accessed 9 May 2015].

[14] Perry S. Marshall, John S. Rinaldi , How to Plan, Install and Maintain TCP/IP Ethernet Networks, ISA, 2004 ISBN 1-55617-869-7.

[15] Hans Berger: Automating with STEP, Function Manual, 2007, ISBN 3-89578-253-X

[16] JOHN, Karl-Heinz; TIEGELKAMP, Michael. IEC 61131-3: programming industrial automation systems: Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making Aids . 2nd ed. New York : Springer, 2010, 390 s, ISBN 978-3-642-12014-5.

[17] SIMOTION SCOUT, Configuration Manual, Edition 2007.

[18] SIMOTION C, Operating Instructions, Edition 2008.

[19] SINAMICS DRIVE CLiQ, [Online]. Available: https://www.industry.usa.siemens.com/drives/us/en/electric-drives/ac-drives/high-performance-and-servo-drives/drive-cliq-motor-drive-integration-system/Documents/DRV-Drive_CLiQ_Brochure.pdf [Accessed 9 May 2015].

[20] SIMOTION ST Structured Text, Programming and Operating Manual, Edition 2008.

[21] SIMOTION Standard Function for Active Line Modules (ALM), Function Manual, Edition 2008.

[22] The structure of the distributed process control system [Online]. Available: http://www.teh-lib.ru/atpip/struktura-raspredeljonnoj-asu-tp/Vse-stranitsy.html [Accessed 10 May 2015].

[23] SIMOTION, Motion Control Path Interpolation, Function Manual, Edition 2014.

CD ROM

The CD ROM contains one ZIP file with following components:

1. PDF file of the diploma thesis;

2. HTML file of the simulation script;

3. ZIP file of the SIMOTION SCOUT project;

4. TXT file of the ST code (Appendix 1);

5. TXT file of the JavaScript code (Appendix 2);

6. AVI file of the demonstration of system work.