sciendo

Management Systems in Production Engineering

2019, Volume 27, Issue 1, pp. 55-62

Date of submission of the article to the Editor: 07/2018
Date of acceptance of the article by the Editor: 11/2018

DOI 10.1515/mspe-2019-0010

THE USE OF LOW-BUDGET SELF-ASSEMBLY SETS FOR RESEARCH AND ROBOTICS EDUCATION

Piotr CHELUSZKA
Silesian University of Technology

Abstract:
The article presents the possibilities of using easily accessible and inexpensive educational sets in scientific research and the process of robotics education. Such kits allow the exploration of theoretical and practical knowledge taking into account aspects of engineering, such as: mechanics, drive systems, sensor systems, control and programming of robots. Models of robots built from inexpensive components can also be used to test new solutions in the field of construction or control algorithms before they are used in real applications. As an example, the model of the palletizing manipulator for self-assembly was shown, the control of which was based on the Arduino Uno controller, while the drives were implemented using low-cost hobby-grade servos. For the kinematic structure of this manipulator, the forward and inverse kinematics task for the position has been discussed. This constituted the basis for the development of a manual control algorithm implemented in the controller – using a joystick and programmed – based on the data sent to the controller using serial communication from a PC. The article presents the results of the computer simulation of the manipulator kinematics, the hardware and software implementation of the robot model and the effects of its operation. The possibility of expanding the control system with additional elements to increase its functionality was indicated.

Key words: *industrial robot, manipulator, self-assembly set, kinematics, control, programming*

INTRODUCTION

In the last decades, robotics is one of the most dynamically developing fields of science and technology. This is evidenced, for instance, by statistics published annually by the International Federation of Robotics (IFR), as well as observations of the world around us. Robotics is not only associated with industrial applications, but more and more often it enters into everyday life of every human being. Intensive growth of robotisation can be observed in industrial applications in many countries of the world, especially in Asia. It applies both to the systematic increase in the number of robots brought into service, and, as a result, increasing robotisation density, understood as a number of robots per 10,000 employees. For example, in 2016, a number of robots working in the industry in the whole world amounted to 1.8 million and was 12% higher than in 2015 (Fig. 1). It is estimated, that by 2020 this number will have almost doubled and exceeded 3 million [1].

Due to the development of robotics and a wider area of robot applications (including agriculture, medicine, logistics, defense, public services, or households) it is necessary to educate engineering personnel for the needs of design, production, technical service and use of robots in industry and beyond. From an early age, children and young students must be involved in activities associated with broadly understood technology, manual skills, and later – an ability to solve theoretical and practical issues underlying design and construction of robots for various purposes. Stimulation of creative thinking from the early days will bring innovative solutions necessary for development of technology – economy – civilization. Modern technology gives a wide range of possibilities to achieve the above–mentioned goal. Widespread availability of inexpensive, but often very technologically advanced elements (especially in the field of electronics), offers wide possibilities of developing both simple and very complex automation systems and robots at home. One of the examples can be homemade robots, like segways [2]. From a very young age, children are given an opportunity to learn about the secrets of robotics, in form of, for example, Lego MindStorms series [3].

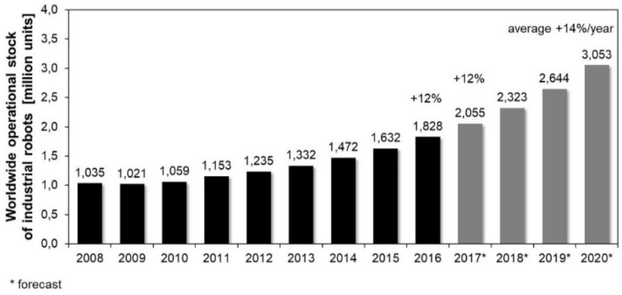

Fig. 1 World operational resources of industrial robots in 2008-2016 and their estimated size in 2017-2020
Source: [1].

Interest in using robot models for educational and research purposes is widening [[4], 5, 6, 7, 8, 9, 10, [11]]. The use of cheap, small educational robots with functional features of robots used in industry does not require large investments and appropriate technical facilities.

This article presents possibilities of using the "Robot Arm" educational set (Fig. 2) offered by Mageek [12] for the purpose of demonstrating the basic issues of robotics.

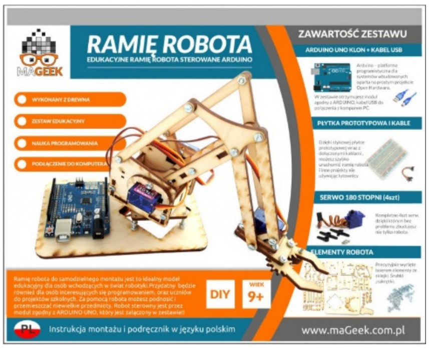

Fig. 2 Educational kit for self-assembly Mageek "Robot arm"
Source: [12].

It is a set from the group of DIY (*Do it yourself*) products and, therefore, it is intended for self–assembly, which requires only a screwdriver. The set, costing around PLN 160, includes:

- structural elements of the industrial robot manipulator model made of modeling plywood in laser cutting technology,
- fasteners (a set of screws with different lengths and nuts),
- Arduino Uno R3 driver – AVR clone controller,
- four TowerPro SG-90 servo drives,
- prototyping board,
- electrical wires to make the necessary electrical connections.

An included manual shows subsequent stages of manipulator assembly. It also shows a user the basics of servo drive control, using the Arduino Uno driver and its programming. The paper discusses hardware management of robot's control system and basic issues from the scope of kinematics of the manipulator under consideration, constituting a starting point for programming the controller for performing motion according to the adopted control strategy.

HARDWARE MANAGEMENT

The example under discussion concerns a model of an industrial palletizing robot, consisting of a manipulator (Fig. 3a), a controller (Fig. 3b) and a power supply system. Due to a kinematic structure, which will be discussed in Chapter 3, the manipulator is equipped with four ModelPro SG-90 modeling servomechanisms: three for driving individual axles of the manipulator and one for driving a gripper. Manipulator drives are controlled with Arduino Uno R3 board – AVR clone. Due to the fact that the robot in controlled in two ways – manually with a joystick or using a software (based on data sent from the PC to the controller), the above-described educational set has been extended with additional elements:

- a two-direction joystick,
- Wi-Fi NodeMCU module (ESP8266-12E),
- a logic level converter,
- a set of basic electronic components, such as: LED diodes, resistors, mounting potentiometers, a micro switch and an additional prototyping board.

The total cost of the above-mentioned items was PLN 100. Therefore, the total cost of purchasing elements for implementation of this project did not exceed PLN 260.

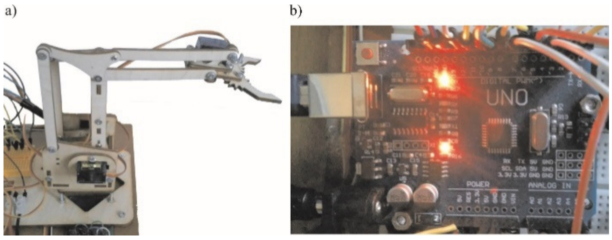

Fig. 3 Palletizing robot model:
a) manipulator, b) Arduino Uno R3 controller

Figure 4 shows a schematic diagram of the Arduino Uno controller connections with manipulator drives and other elements used in the discussed project. In order to adjust an exit position (parking) of the manipulator to the analog ports, there are four mounting potentiometers (P1-P4) connected, to enable adjusting the manipulator drives (trimmer), activated with a S1 switch. The gripper is controlled manually with S2 and S3 keys. Gripper status (open/closed) is indicated by LED diodes (G and R). Servo drives are controlled from digital ports (No. 6 and 9-11) with modulated pulse width (PWM). The control system operates in the open feedback loop, which results from the construction of used servomechanisms. This is why, an issue of positioning accuracy is not considered here.

In case of software control, manipulator position data from a PC can be transmitted in two ways:

- from the COM terminal via the USB port (directly to the Arduino Uno controller),
- wirelessly – using the Wi-Fi module.

The Wi-Fi NodeMCU module is connected to the Arudino Uno controller via a software serial port (SoftwareSerial). Due to the different voltage levels in Arduino Uno (5 V) logic and NodeMCU (3.3 V), this communication takes place via a 3.3/5 V logic level converter.

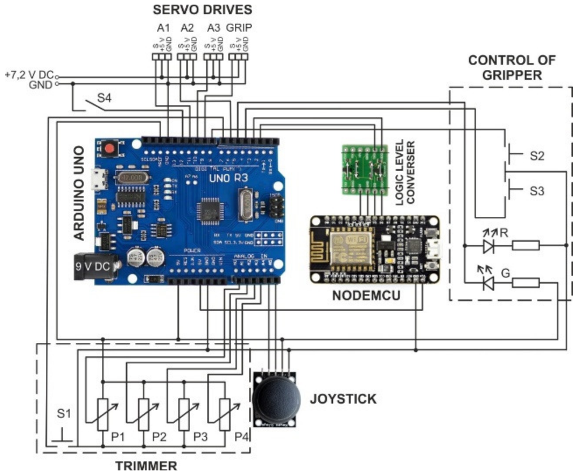

Fig. 4 Schematic diagram of the palletizing robot model control system

MANIPULATOR KINEMATICS

The palletizing manipulator, whose educational model is presented in this paper, belongs to the group of joint manipulators, with a kinematic chain made only from rotational pairs. Due to the nature of the tasks performed by this kind of industrial robots, the tool (gripper) is always set parallel to the plane of the manipulator's base. This can be ensured, for example, by a simplified system formed by two articulated quadrilaterals built on the base of the arm and forearm (Fig. 5). Real manipulators of this type usually have four DOF (they are equipped with four independent drives: a shoulder, arm, forearm and, additionally, a wrist drive, thanks to which a gripper may rotate about a vertical axis). A discussed manipulator model does not have the last of the possibilities, while gripper jaws are driven by a fourth servomechanism.

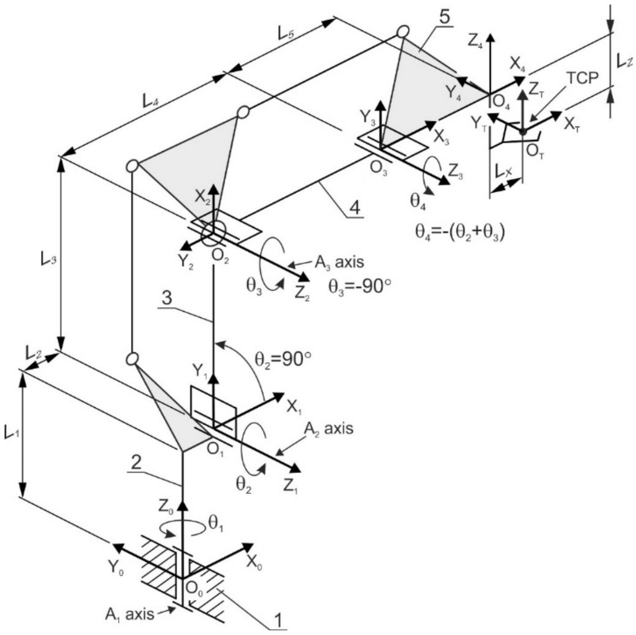

Fig. 5 Kinematic diagram of the palletizing manipulator model: 1 – base, 2 – shoulder, 3 – arm, 4 – forearm, 5 – wrist

The base (1), the shoulder (2), the arm (3), the forearm (4) and the wrist (5) are connected with the Cartesian coordinate systems: $X_0Y_0Z_0$, …, $X_4Y_4Z_4$. In turn, the tool (gripper) has been assigned the $X_TY_TZ_T$ coordinate system hooked to the tool central point (TCP). The Denavita-Hartenberg (D-H) notation is used to describe a position and orientation of the manipulator elements [13]. Table 1 presents the D-H parameters and the values of geometrical quantities of the analyzed manipulator model.

Table 1

A set of Denavit–Hartenberg parameters for the considered palletizing robot manipulator

No. of link	a_i	α_i	d_i	q_i	[mm]	
1	L_2	90°	L_1	θ_1	$L_1 = 55$	$L_5 = 23$
2	L_3	0	0	θ_2	$L_2 = 13$	$L_X = 34$
3	L_4	0	0	θ_3	$L_3 = 80$	$L_Z = 6$
4	L_5	-90°	*0*	$\theta_4 = -(\theta_2+\theta_3)$	$L_4 = 80$	

Because we are dealing with rotational pairs, configuration variables (joint space) are angles θ_1, …, θ_4, while the value of the latter depends on the current values of the arm inclination angle (θ_2) and the forearm (θ_3). Position of the TCP (beginning of the $X_TY_TZ_T$ coordinate system) in the basic coordinate system ($X_0Y_0Z_0$) can be determined based on transformation of coordinate systems:

$$\begin{bmatrix} {}^0x_{OT} \\ {}^0y_{OT} \\ {}^0z_{OT} \\ 1 \end{bmatrix} = {}^0\boldsymbol{T_4}(\theta_1,\theta_2,\theta_3,\theta_4) \cdot \begin{bmatrix} L_X \\ 0 \\ -L_Z \\ 1 \end{bmatrix} \tag{1}$$

where:

${}^0x_{OT}$, ${}^0y_{OT}$, ${}^0z_{OT}$ – coordinates of the TCP in the basic system,

${}^0\mathbf{T_4}(\theta_1, \theta_2, \theta_3, \theta_4)$ – complex homogeneous transformation matrix:

$${}^0\boldsymbol{T_4}(\theta_1,\theta_2,\theta_3,\theta_4) = \boldsymbol{H_1}(\theta_1) \cdot \boldsymbol{H_2}(\theta_2) \cdot \boldsymbol{H_3}(\theta_3) \cdot \boldsymbol{H_4}(\theta_4) \tag{2}$$

Homogeneous transformation matrices $\mathbf{H_1}(\theta_1)$, …, $\mathbf{H_4}(\theta_4)$ are at the same the product of homogeneous matrices describing translations and rotations necessary to carry out transition between successive coordinate systems. These matrices have the following form:

$$\boldsymbol{H}_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cdot\cos\alpha_i & \sin\theta_i\cdot\sin\alpha_i & a_i\cdot\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cdot\cos\alpha_i & -\cos\theta_i\cdot\sin\alpha_i & a_i\cdot\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

for i = 1, 2, 3, 4

Elements of a complex homogeneous transformation matrix ${}^0\mathbf{T_4}$ with the dimension 4×4 determine position of a starting point and orientation of the axis of the coordinate system $X_4Y_4Z_4$ in the basic system $X_0Y_0Z_0$:

$${}^0\boldsymbol{T_4}(\theta_1,\theta_2,\theta_3,\theta_4) = \begin{bmatrix} l_x & m_x & n_x & p_x \\ l_y & m_y & n_y & p_y \\ l_z & m_z & n_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where:

l_x, l_y, l_z, m_x, m_y, m_z, n_x, n_y, n_z – elements of the versor of the axis: X_4, Y_4 and Z_4,

p_x, p_y, p_z, – elements of a radial vector of point O_4.

Tool orientation results directly from manipulator rotation around the Z_0 axis. Axles: Z_4 and Z_T are therefore always parallel to the Z_0 axis, only direction of the X_T and Y_T axles changes as a function of the angle θ_1.

Kinematics tasks for positions will be discussed in the further part of this chapter. Kinematics tasks for speed are not considered, because in the application there is basically no possibility of controlling the manipulator's movement speed. Software control of the speed of the servo drives takes place only by entering the delay with the *delay( )* instruction, stopping the program for the time specified in the argument.

Forward task

A forward task is converting internal position of the manipulator (joint space) to its external position (Cartesian space) [14], [15]. By performing multiplication described by equation (2), after simplifying individual expressions, elements of a complex matrix of homogeneous transformation assume the following uncomplicated form:

$$\begin{cases} l_x = \cos\theta_1 \\ l_y = \sin\theta_1 \\ l_z = 0 \end{cases} \quad \begin{cases} m_x = -\sin\theta_1 \\ m_y = \cos\theta_1 \\ m_z = 0 \end{cases} \quad \begin{cases} n_x = 0 \\ n_y = 0 \\ n_z = 1 \end{cases} \tag{5}$$

$$\begin{cases} p_x = [L_2 + L_3 \cdot \cos\theta_2 + L_4 \cdot \cos(\theta_2 + \theta_3) + L_5] \cdot \cos\theta_1 \\ p_y = [L_2 + L_3 \cdot \cos\theta_2 + L_4 \cdot \cos(\theta_2 + \theta_3) + L_5] \cdot \sin\theta_1 \\ p_z = L_1 + L_3 \cdot \sin\theta_2 + L_4 \cdot \sin(\theta_2 + \theta_3) \end{cases}$$

and the coordinates of the TCP are described with the formulas:

$$\begin{cases} {}^0x_{OT} = L_x \cdot \cos\theta_1\, [L_2 + L_3 \cdot \cos\theta_2 + L_4 \cdot \cos(\theta_2 + \theta_3) + L_5] \cdot \cos\theta_1 \\ {}^0y_{OT} = L_x \cdot \mathrm{SIN}\,\theta_1\, [L_2 + L_3 \cdot \cos\theta_2 + L_4 \cdot \cos(\theta_2 + \theta_3) + L_5] \cdot \sin\theta_1 \\ {}^0z_{OT} = -L_Z + L_1 + L_3 \cdot \sin\theta_2 + L_4 \cdot \sin(\theta_2 + \theta_3) \end{cases} \tag{6}$$

Fig. 6a-6c show the effect of solving a forward kinematics problem for the position of th analyzed manipulator model. For the assumed variability ranges of rotation angles of the shoulder (θ_1 = –60° ÷ +60°), arm inclination (θ_2 = +50° ÷ +130°) and the possible range of forearm inclination (θ_3), subsequent positions of the TCP determine the working range of the manipulator. The set of these points is limited by the surface determining the shape of the working space of the manipulator. Due to constructional reasons (possible collision of the lever of the parallel linking mechanism), the range of variability in the forearm angle is limited and depends on the current value of the inclination angle of the arm (Fig. 6d). The greater the θ_2 angle (the more the arm is tilted back), the smaller the range of variability in the angle of the forearm.

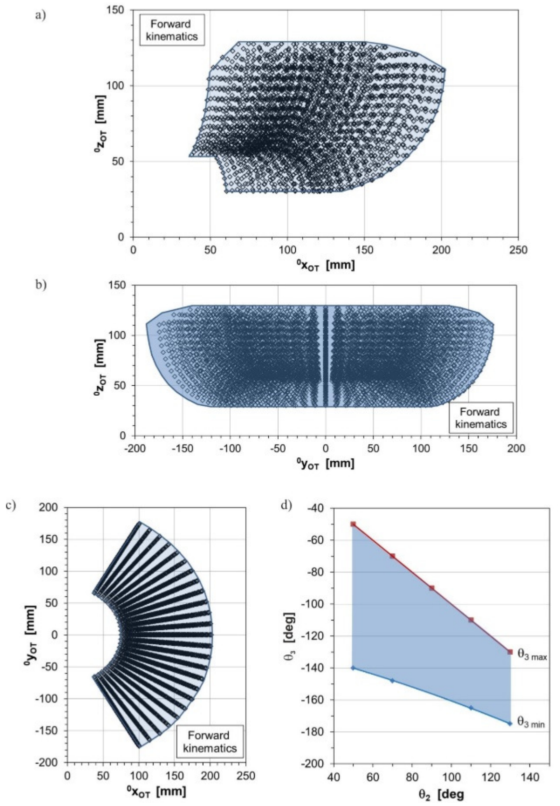

Fig. 6 Working range of the examined palletizing manipulator model as a result of the solution of a forward kinematics task for positions (a -) and the range of the forearm inclination angle depending on the arm inclination angle (d)

Inverse task

The inverse task of kinematics for the position consists in determining the internal position of the manipulator (joint space) on the basis of its external position (Cartesian space). It is limited to determining all possible sets of values of configuration coordinates enabling reaching the desired position and orientation of the tool [14], [15]. The inverse task consists in finding solutions of the system of equations (5), treating of axles of the $X_4Y_4Z_4$ (**l**, **m** and **n**) coordinate system and elements of the radial vector of this system – point O_4 (**p**) as data of the versors. While component values of the vector **p** (p_x, p_y, p_z) result from the values of the coordinates of the TCP in the basic system:

$$\begin{cases} p_x = {}^0x_{OT} - L_x \cdot \cos\theta_1 \\ p_y = {}^0xy_{OT} - L_x \cdot \sin\theta_1 \\ p_z = {}^0z_{OT} + L_z \end{cases} \tag{7}$$

In this case, only three of 12 equations are independent. So we can build a system of three equations with three unknowns (θ_1, θ_2, θ_3) of the form:

$$\begin{gathered} [L_2 + L_3 \cdot \cos\theta_2 + L_4 \cdot \cos(\theta_2 + \theta_3) + L_5] \cdot cos\theta_1 - p_x = 0 \\ L_1 + L_3 \cdot \sin\theta_2 + L_4 \cdot \sin(\theta_2 + \theta_3) - p_z = 0 \\ \cos\theta_1 - m_y = 0 \end{gathered} \tag{8}$$

As a result of solving a system of equations (8), we obtain formulas enabling determination of the manipulator configuration coordinate values, when the location and orientation of the tool coordinate system is given:

$$\begin{cases} \theta_1 = arctan\left(\dfrac{{}^0y_{OT}}{{}^0x_{OT}}\right) \\ \theta_2 = \begin{cases} 2arctan\left(\dfrac{-A - \sqrt{A^2 + B^2 - C^2}}{C - B}\right), when\ c \neq b \\ \dfrac{\pi}{2}, when\ C = B \end{cases} \\ \theta_3 = arcsin\left(\dfrac{{}^0z_{OT} + L_Z - L_1 - L_3 \cdot \sin\theta_2}{L_4}\right) - \theta_2 \\ \theta_4 = -(\theta_2 + \theta_3) \end{cases} \tag{9}$$

while:

$$\begin{gathered} A = 2 \cdot L_3 \cdot \left({}^0z_{OT} + L_z - L_1 \right) \\ B = 2 \cdot L_3 \cdot \left(\frac{{}^0x_{OT}}{\cos\theta_1} - L_x - L_2 - L_5 \right) \\ C = L_4^2 - L_3^2 - \left({}^0z_{OT} + L_z - L_1 \right)^2 - \left(\frac{{}^0x_{OT}}{\cos\theta_1} - L_x - L_2 - L_5 \right)^2 \end{gathered} \tag{10}$$

Solving a system of equations (8) leads to a quadratic equation with one unknown (θ_2). Only one of two solutions of this equation is, however, justified. Since the following condition must be met:

$$\theta_2 > 0 \cap \theta_3 < 0 \tag{11}$$

The condition (11) is met in the solution of the following formula (9).

The effect of solving the inverse kinematics problem for the manipulator model considered in this paper is shown in the Figures: 7-9. The obtained sets of points reflect the values of individual configuration coordinates (values of angles of axis rotation of the manipulator) corresponding to different positions of the TCP in space (Fig. 7 and 8). For a given coordinate value $^0z_{OT}$, the points representing the values of tilting angle of the manipulator elements (θ_1, θ_2, θ_3 and θ_4), corresponding to different positions of the TCP on the plane parallel to the plane of its base (X_0Y_0) create three-dimensional surfaces of different shape (Fig. 9).

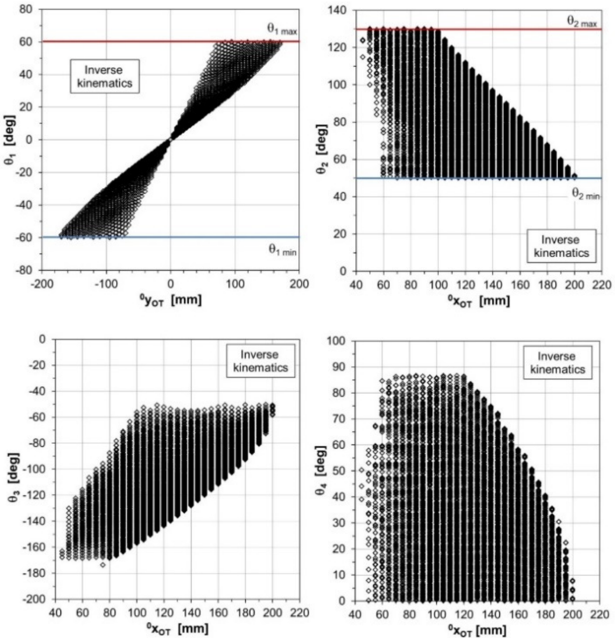

Fig. 7 The effect of solving the inverse kinematics task for positions – a family of points representing coordinate values for different positions of the gripper of the manipulator

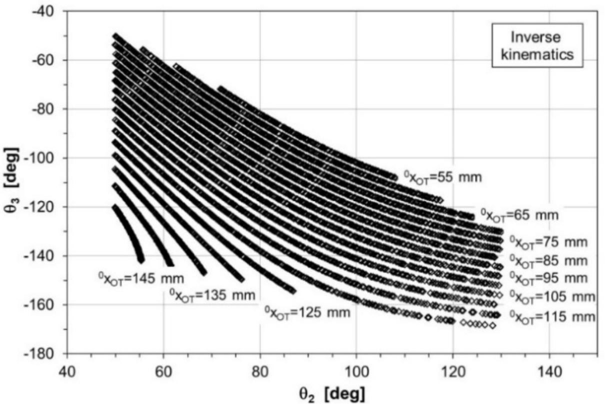

Fig. 8 The dependence of the forearm angle of inclination from the angle of inclination of the arm for different positions of the gripper of the tested manipulator

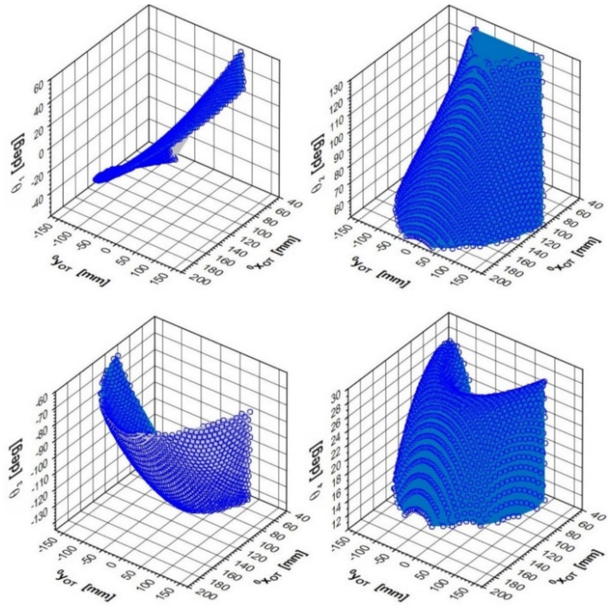

Fig. 9 Dependence of configuration coordinates on the location of a TCP point in a plane parallel to the X_0Y_0 plane for $^0z_{OT}$ = 90 mm

IMPLEMENTATION OF MANIPULATOR KINEMATICS IN THE ROBOT MODEL CONTROLLER

A prepared kinematic model of the palletizing robot manipulator was implemented in the Arduino Uno controller. Arduino 1.8.5 free Integrated Development Environment (IDE), available on the Arduino.cc platform [[15]], was used for programming the controller. After compilation, a program code is loaded into the controller's memory using the serial port.

In the structure of the written program for Arduino controllers, two basic sections should be distinguished (Fig. 10): *setup( )* and *loop( )*. The first one is called after starting the program. It is used to initiate variables, give them initial values, establish communication and perform other operations required by the user (in this case, for example, initiation of servo drives and reading of the trimming potentiometer settings). The second one is an infinite loop containing instructions executed during each call. In addition to the above-mentioned functions, the program consists of a series of procedures and functions called from attached libraries and placed in the program source code. The program developed for controlling the manipulator of the palletizing robot includes, among others, the following procedures:

- solving forward and inverse kinematics tasks for positions,
- adjusting the position of the manipulator axis (trimming),
- receiving and decoding data,
- manual control of the manipulator with a joystick.

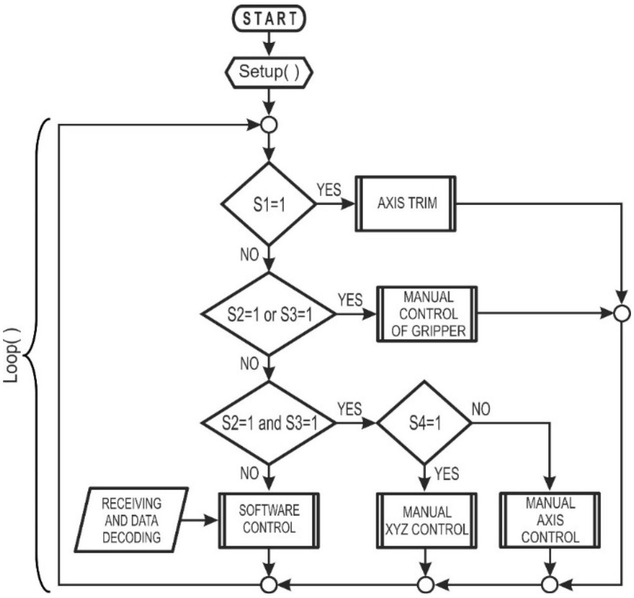

Fig. 10 Algorithm of the control procedure of the palletizing robot manipulator model

The *loop( )* reads the input statuses, including the position of the switches (S1-S4), which determine introduction of the relevant code fragments that enable:

- adjusting the position of the manipulator axis with potentiometers P1-P4 (switch S1 = 1 – pressed),
- manual control of the gripper (switch S2 = 1 – pressed → gripper open or switch S3 = 1 pressed → gripper closed),

- manual manipulation of the manipulator with a joystick (switches: S2 = 1 and S3 = 1 – pressed simultaneously),
- selection of manual control mode – individual manipulator axles or movement in a Cartesian reference system (switch S4 = 1 – pressed),
- reading and decoding of the manipulator predefined position data and automatic control, according to the predefined control strategy (PTP or with linear interpolation).

A controlling mode (individual axles separately or in the Cartesian reference system) determines a steps taken during development of control signals for the servo drives. In the first case (axis control), it is necessary to enter values of coordinate configuration coordinates (angles of individual axles rotation) which must be achieved while moving the manipulator. The control system changes values of these angles with the specified step in the loop until the position is reached. For current manipulator setting, its external position is determined as a result of solving a forward kinematics task. Such a procedure is carried out both in manual control and during PTP software control. In case of manual control in a Cartesian system, data are predetermined values that determine a target position of the TCP in the basic system $X_0Y_0Z_0$. In the next steps, the inverse kinematics task is solved, based on which the current values of rotation angles of the individual manipulator axles are determined. These data are then performed by servo drives until reaching the target position. This algorithm is performed both during manual control and during software control with linear interpolation.

As mentioned before, further manipulator positions can be determined manually with a joystick or with software – as a result of sending a proper set of data to the control system (Fig. 11).

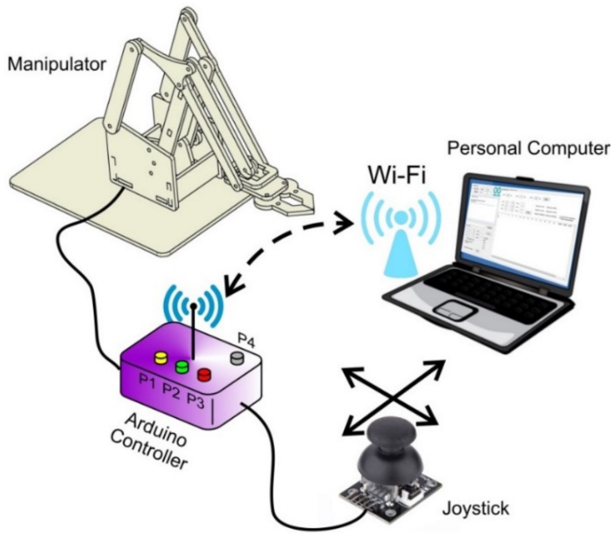

Fig. 11 Ways of setting the position of the manipulator – manually by means of a joystick or programmed – from a PC via WLAN

After decoding, the data is then performed by the manipulator drives. In the first case, the manipulator movement is based on the values read out on the analog inputs of the Arduino controller. In the second case, the controller receives data sets in the form of 26-byte words with the following structure:

AAA:±XXX.X,±YYY.Y,±ZZZ.Z,C

where:

AAA – positioning instruction identifier (KAT – individual manipulator axles control, PTP – *Point-To-Point* control, LIN – linear interpolation),

XXX.X, …, **ZZZ.Z** – predetermined values: rotation angles of the manipulator axis (for AAA=KAT) or coordinates of the TCP in the $X_0Y_0Z_0$ system (for AAA = PTP or LIN),

C – gripper status (C = 0 – gripper open, C = 1 – gripper closed).

Figures 12 and 13 show an example of the results of tests performed on an actual object (palletizing robot manipulator model), behaviour of the manipulator being tested during manual control with a joystick in a Cartesian system. Solving the inverse kinematics task led to recording a course of variability in coordinates of the gripper's TCP (Fig. 12a) and corresponding configuration coordinates of the manipulator (Fig. 12b). As can be seen, movement of the tool along the basic axis of the reference system results from connecting rotational movements of all three manipulator axles. This effect is clearly visible after overlapping subsequent positions of the manipulator while moving the tool along the axis of the $X_0Y_0Z_0$ coordinate system (Fig. 13).

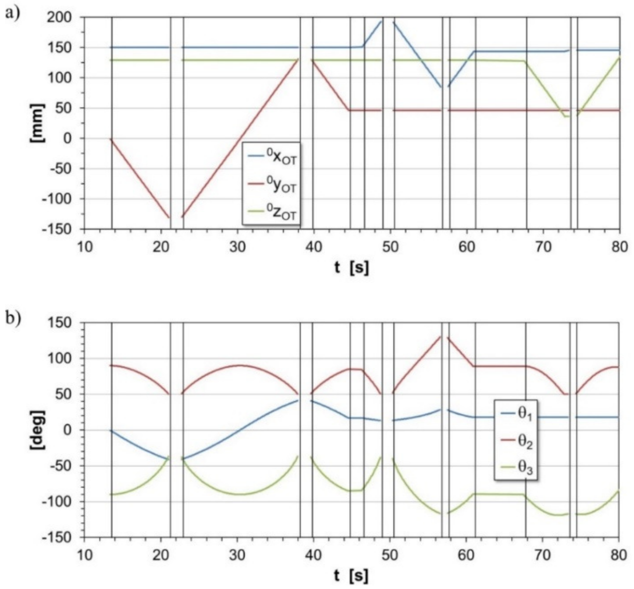

Fig. 12 Manual control in the Cartesian reference system – time courses:

a) coordinates of the TCP point, b) angles of rotation of the individual axis of the manipulator as the effect of solving the inverse kinematics task for the position

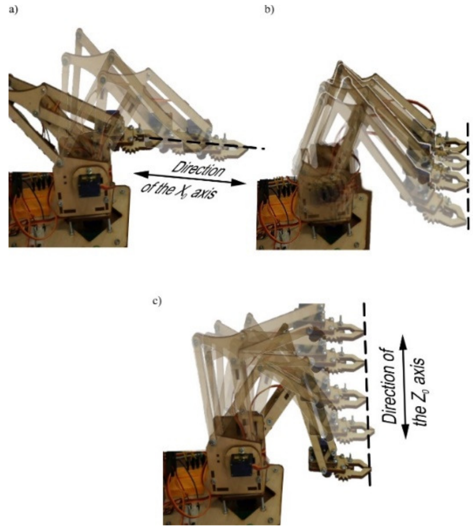
a)
Direction of the X_0 axis
b)
Direction of the Y_0 axis
c)
Direction of the Z_0 axis

Fig. 13 Movement of the manipulator during manual control along:
a) X_0 axis, b) Y_0 axis, c) Z_0 axis of the basic reference system

The final solution of the presented didactic model is a possibility of executing manipulator movement with software manner, where a defining device (PC) sends a sequence of positioning instructions (data sets describing subsequent positions of the tool and how to achieve them), and the control system properly controls the manipulator drives. An example of the movement of a manipulator with linear interpolation based on data sent to the controller via a WLAN is shown in Figure 14. Within about 3 seconds, the TCP is moved from the starting point P_1 (150, 0, 35) to the final point P_2 (70, 100, 120) along a straight line (Fig. 14a, b). This is the effect of the rotational movement of individual manipulator components controlled in an appropriate way. The rotational angle of the shoulder (θ_1) changes approximately linearly in the range from zero to 55° (Fig. 14c – blue line). At the same time, the arm inclination angle (θ_2) rises from 50° to the maximum value (112°), and then slightly decreases (the red line). The value of the forearm inclination angle (θ_3) varies between –120° and –138°, and then increases to –114° when the tool reaches the target position (green line).

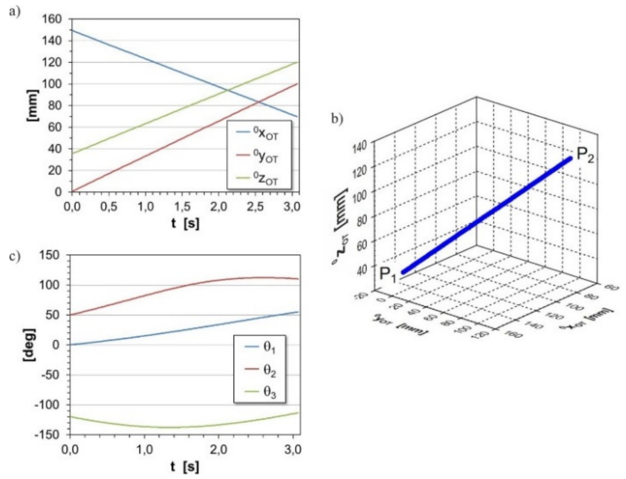

Fig. 14 Programmed control – linear interpolation:
a) time courses of coordinates of a TCP point, b) TCP point path, c) time courses of angles of rotation of individual manipulator axes as a result of solving the inverse kinematics task for positions

SUMMARY

A wide range of, more or less, technically advanced self–assembly kits may be an interesting didactic aid in educating young people in the area of academic subjects and future robotics engineers. This field requires comprehensive technical knowledge, necessary for understanding the essence of robot operation and constituting the basis for creating innovative, increasingly sophisticated solutions. Widespread availability of relatively cheap sets and elements of drive technology, controlling and sensor technology, as well as availability of free tools for creating controller software provides the opportunity to build robotic systems with a more or less complex structure at home. A wide offer makes it possible to stimulate technical interests in the field of mechanical and electrical engineering, electronics and computer science even in pre-school and school children. For example, the educational set discussed in this paper is intended for people aged 9 and older. Building educational robots (and not only) from ready-made elements, as well as increasing availability of, for example, 3D printing techniques, provide a wide range of possibilities of developing own solutions limited only by the imagination of their creators.

The use of low-budget kits in education from an early age is a factor that stimulates curiosity of learning about broadly understood technology. This is an excellent starting point in education of engineering and technical personnel in the field of robotics, characterized by constantly growing demand for highly qualified employees. According to the literature on the subject, robot models are also a convenient aid in scientific experiments and research and development works in the field of robotics.

An above–discussed example of a toy manipulation robot used to explain theoretical bases of operation of manipulative industrial robots and their control and programming, is not a closed project. It can be developed further with new functionalities and modified. This is a common feature of these types of sets. They are not finished projects, but designed to create the user's own creative solutions.

REFERENCES

[1] *World Robotics – Industrial Robots 2017*, Report of International Federation of Robotics – Statistical Department. Frankfurt/Main, Sep. 2017.

[2] http://www.instructables.com/id/Arduino-Segway/ [Apr. 26, 2018].

[3] https://www.lego.com/pl–pl/mindstorms [Apr. 26, 2018].

[4] M. Abu Qassem, I.M. Abuhadrous, H. Elaydi. “Modeling and Simulation of 5 DOF educational robot arm”, in *Proc. of 2nd International Conference on Advanced Computer Control (ICACC)*, vol. 5, 2010, pp. 569–574, DOI 10.1109/ICACC.2010.5487136.

[5] B.E. Byambasuren, D. Kim, M. Oyun-Erdene, C. Bold, J. Yura. “Inspection robot based mobile sensing and power line tracking for smart grid”. *Sensors*, vol. 16, iss. 2, no. 250, 2016, pp. 1-14, DOI:10.3390/s16020250.

[6] M. Górska, M. Olszewski. "Brain-computer interface in the task of mobile robot control". *PAR magazine*, vol. 19, no. 3, 2015, pp. 15-24, DOI: 10.14313/PAR_217/15.

[7] M. Leba, E. Pop. "Articulated robotic arm simulation and control". *Annals of the University of Petrosani, Mechanical Engineering*, no. 8, 2006, pp. 181-188.

[8] K. Murawski, J. Chudzikiewicz, J. Turczyn. „Mobilny robot edukacyjny KMURAW-3 – rozwiązania techniczne sprzętu i oprogramowania". *Biuletyn Instytutu Automatyki i Robotyki*, nr 21, 2004, pp. 31-44.

[9] R. Neerparaj, G.B. Palzor, P. Udit. "Remote learning: android operated educational robot arm with 6 DOF". *International Journal of Electrical, Electronics and Data Communication*, vol. 2, iss. 11, 2014, pp. 58-61.

[10] D. Rivas, M. Alvarez, P. Velasco, J. Mamarandi, J.L. Carrillo-Medina, V. Bautista, O. Galarza, P. Reyes, M. Erazo, M. Pérez, M. Huerta. "BRACON: Control system for a robotic arm with 6 degrees of freedom for education systems", in *Proc. of the 6th International Conference on Automation, Robotics and Applications*, Feb 17-19, 2015, Queenstown, New Zealand, pp. 358-363.

[11] M. Węgierek, B. Świstak, T. Winiarski. „Modularized environment for Line Follower robots". *PAR magazine*, vol. 19, no. 3, 2015, pp. 61-66, DOI: 10.14313/PAR_217/61.

[12] https://mageek.com.pl/ [Apr. 26, 2018].

[13] J.J. Craig: *Introduction to robotics: mechanics and control*. Pearson Education International, 2005.

[14] S. Kucuk, Z. Bingul. "Robot Kinematics: Forward and Inverse Kinematics", in *Industrial-Robotics-Theory-Modelling-Control*, S. Cubero (ed.), ARS/PLV, 2006, pp. 117-148.

[15] Morecki, J. Knapczyk (red.): *Podstawy robotyki. Teoria i elementy manipulatorów i robotów*. WNT, 1999.

[16] https://www.arduino.cc [Apr. 26, 2018].

Assoc. Prof. Piotr Cheluszka, PhD. DSc. Eng.
Silesian University of Technology,
Faculty of Mining and Geology,
Department of Mining Mechanization and Robotisation
ul. Akademicka 2, 44-100 Gliwice, Poland
e-mail: piotr.cheluszka@polsl.pl