ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2024. №4 (30) ISSN 2524-2296 (online)

UDC 004.896 DOI: https://doi.org/10.30837/2522-9818.2024.4.005

A. BRONNIKOV, M. BENDEBERYA

DEVELOPMENT OF A MANIPULATOR KINEMATIC MODEL USING ABB ROBOT STUDIO

The **subject** of the research in the article is the application of control technology and algorithms, construction of the trajectory of movements and interaction with the production environment based on Abb Robot Studio. The **goal** of the work is the simulation of a kinematic model of the manipulator, which will allow effective performance of actions in real time, using the mathematical calculations and the Abb Robot Studio software. The article addresses the following **tasks**: to analyze the current state of integration of manipulators into production processes, as well as to choose and substantiate the development platform and robot model, to develop a kinematic model and design and write software for controlling manipulators based on mathematical calculations. The use of mathematical analysis **methods** and the theory of automatic control to simulate the operation of an industrial robot, simulation methods – to check the operation of an industrial robot model in the ABB ROBOT STUDIO environment. The following **results** were obtained: an analysis of the current state of integration of manipulators into production processes, in particular in the era of Industry 4.0, was carried out, a manipulative robot was selected and substantiated, which satisfies the mathematical calculations carried out, a kinematic model was developed, and its design was carried out, software for controlling manipulators was developed based on the conducted mathematical calculations. **Conclusions**: a model and software module were developed for the adaptive execution of the production process as part of the virtual model of the ABB IRB 1200 robot, which allows to increase the efficiency of the production process, reduce downtime, and improve product quality. The optimization of trajectories and collision management allowed to reduce the maintenance and operation costs of the robot, which led to an overall reduction in production costs, which will also be reflected in long-term use. The developed software module allows the work to quickly adapt to changes in the production process, which provides high flexibility and the ability to quickly adjust to new tasks.

Ключові слова: Industry 4.0, Abb Robot Studio, manipulator, robot, production, kinematic model, mathematical calculations.

Introduction

The rapid evolution of production processes is driven by various factors, and the adoption of automation and robotic systems plays a crucial role. Some of the key aspects of Industry 4.0 are [1–2]:

– technological advancements: the increasing power of computers and advancements in specialized software have enabled the creation of more sophisticated and efficient robotic systems. Sensors and artificial intelligence algorithms allow robots to perceive their environment, analyze data, and adapt to specific tasks effectively;

– economic considerations: enterprises strive to remain competitive, which compels them to adopt cutting-edge technologies, including robotics. Over time, the cost of robots and associated equipment has decreased significantly, making them accessible to a broader range of businesses. Implementing robotic systems can lead to cost savings and increased productivity.

– gradual improvement: enterprises often seek process improvements beyond the initial setup of production. Thoughts about automation or enhancing specific components emerge as the need to optimize efficiency and expand capabilities arises.

Installing automated areas without disrupting ongoing production is essential. This involves careful planning and integration.

The convergence of technological advancements, economic factors, and the desire for continuous improvement drives the adoption of robotic systems in production. By strategically implementing automation, businesses can enhance efficiency while maintaining control over the production process.

1. Manipulation robotics and its control as a part of manufacturing

There are a lot of publications, connected with related topic. For example, [3] describes kinematic model optimization and self-collision detection technology for a mobile manipulator, [4] considers various methods of kinematic modeling of robotic manipulators, [5] describes the basic principles of creating a kinematic model of a manipulator, [6] describes the design, modeling, and kinematics analysis of a modular cable manipulator, [7] is an overview of modern procedures of kinematic design of manipulators, [8] describes the modeling and control of a six-domed robotic

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2024. No. 4 (30)

manipulator, [9] shows analysis of the work space of a four-house robotic manipulator, [10–11] describes the kinematic modeling of a robot.

In modern manufacturing environments, industrial robots play a pivotal role in various tasks, including:

– assembly: robots can precisely assemble components, ensuring consistent quality and reducing human error;

– material handling: they efficiently move materials within production lines, warehouses, and logistics centers;

– packaging: robots automate packaging processes, improving speed and accuracy;

– welding and cutting: ABB robots excel in welding and cutting applications, enhancing productivity.

ABB, a leading manufacturer of industrial robots, has significantly transformed production across industries [12]:

– productivity: ABB robots work tirelessly, leading to increased output and reduced cycle times;

– quality: their precision ensures consistent product quality, minimizing defects;

– safety: by handling hazardous tasks, robots enhance workplace safety for human workers;

– flexibility: ABB robots adapt to changing production conditions, allowing rapid adjustments.

ABB's manipulative robots owe their capabilities to cutting-edge technologies and use sensors. These devices provide critical information about the robot's environment. For instance:

– position sensors: determine the robot's location and orientation;

– force/torque sensors: enable delicate handling of objects;

– vision sensors: allow robots to recognize and manipulate items accurately.

Artificial Intelligence (AI) application in robotics are:

– ABB robots leverage AI algorithms for real-time decision-making;

– they learn from experience, adapting to varying conditions;

– AI-driven path planning ensures efficient movement and collision avoidance.

D. Raina and S. K. Saha rightly emphasizes that robots empower manufacturers to respond swiftly to market demands while maintaining high-quality standards. As sensor technologies and AI continue to evolve, manipulative robots will play an even more crucial role in shaping the future of manufacturing [13].

There are the various methods of controlling robots:

a) uncontrolled method: in this method, the working body of the machine is set in motion without any specific control. It operates freely based on its inherent dynamics. Examples include simple mechanical systems like pendulums or spinning tops.

b) controlled method: controlled robots have a more deliberate approach to motion. These controlled robots are commonly used in industrial applications, where precise movement is essential. Here are two variations:

– constant speed control: the driven working body operates at a fixed speed.

– variable speed control: the robot adjusts its speed based on the task requirements.

c) program-controlled method: in program-controlled robots, an external program specifies the robot's actions. The program defines the sequence of movements, positions, and tasks. These robots follow a predetermined set of instructions, making them suitable for repetitive tasks.

d) follower method: follower robots automatically track a reference signal (usually from an external source). They maintain a certain accuracy in replicating the movement of the reference signal. Examples include robotic arms following a conveyor belt or a moving assembly line.

e) adaptive method: adaptive robots dynamically adjust their parameters based on changing operating conditions. These parameters might include speed, force, or other control variables. Adaptive control ensures optimal performance even when the environment or task changes.

The use of software-controlled methods in robotics is indeed crucial for achieving accuracy, flexibility, automation, and seamless integration within production processes.

The importance of software-controlled methods can be determined in such abilities:

– accuracy: software-controlled robots can precisely execute predefined tasks, ensuring consistent results. Whether it's assembling components or handling materials, accuracy is vital;

– flexibility: software-based control allows robots to adapt to changing conditions. You can reprogram them easily for different tasks without major hardware modifications;

– automation: by automating repetitive tasks, software-controlled robots free up human resources and improve overall efficiency;

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2024. №4 (30) ISSN 2524-2296 (online)

– integration: seamless integration with existing systems streamlines production workflows.

But there are several challenges with robotic integration in modern manufacturing:

– not all robots fit seamlessly into established processes.

– this integration often requires additional resources, energy, and time. Customization becomes essential.

– the market for flexible robotic solutions can be narrow, but it's essential to find systems that align with specific production needs.

ABB RobotStudio is a powerful platform for offline programming, simulation, and optimization of robots. The key features of this program are:

– offline programming: users can create and test robotics programs on a computer without interrupting production. Direct communication with real robots allows efficient program development.

– 3D visualization and simulation: RobotStudio provides high-quality 3D visualization. Users can see how robots will perform tasks, identify issues, and make corrections before actual production. This virtual testing minimizes risks and ensures smoother implementation.

– trajectory optimization: determining efficient movement routes is critical for productivity. RobotStudio helps optimize robot paths, reducing task completion time. Especially valuable in complex processes where every second matters.

– virtual tools and sensors: virtual sensors and grippers allow realistic simulation. Online controllers visualize the robot's behavior without physical components. Debugging and fine-tuning become easier.

– AR visualization and integration: augmented reality (AR) models aid in accurate integration. Elements can be virtually projected and adjusted for seamless process alignment.

ABB RobotStudio empowers users to create, test, and enhance robotics programs efficiently. It's a valuable tool for achieving precision, safety, and productivity in robotic operations.

2. Software and robot model justification for further development

ABB Robotics is renowned for its cutting-edge robotics and machine automation systems. As a leading supplier, ABB offers a comprehensive and integrated portfolio that covers a wide range of applications, including robots, autonomous mobile robots (AMRs), and machine automation solutions. These systems are meticulously designed and orchestrated through specialized software.

ABB's software allows precise adjustments to the robot's behavior based on the specific operating environment. This adaptability ensures optimal performance and safety.

By simulating the robot's behavior virtually, developers can fine-tune parameters without physically interacting with the robot. This minimizes risks during experimentation and system improvement [14].

Developers can create experimental scenarios within the virtual environment provided by ABB's software.

This approach enables them to explore the robot's capabilities, test control strategies, and evaluate performance without affecting the physical robot.

The ability to conduct experiments without direct interference allows for iterative development.

As the system evolves, developers can refine algorithms, optimize settings, and enhance performance seamlessly.

RobotStudio is ABB's powerful software tool for offline robot simulation. It offers:

– offline programming and simulation;

– training and optimization;

– integration with real models.

RobotStudio allows integration with any ABB robot model available in real time. For analysis, were chosen the smallest initial models: IRB 1090, IRB 1100, and IRB 1200.

Examining the IRB 1200 robot model, we can highlight several key features. Firstly, this robot is versatile and can handle a wide range of applications. Additionally, it can be installed at various angles. The design also incorporates safety considerations for working with food products. Notably, the IRB 1200 lacks a second axis displacement, resulting in a longer stroke compared to other small robots. This design choice allows the robot to be positioned closer to products, leading to a more compact installation.

Fig. 1 illustrates the robot's working range, revealing that it is fully reversible along both the x-axis and z-axis (with some exceptions in specific zones of axis change or rotation) and table 1 shows robot features.

When developing a robot [15], careful consideration of its future application environment is crucial, as it directly impacts its overall form. Decisions about the robot's purpose should be made early in the design process. Based on this defined framework, structural elements can be added as needed. Therefore, thorough material research and practical exploration of the main components are essential.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Innovative technologies and scientific solutions for industries. 2024. No. 4 (30)

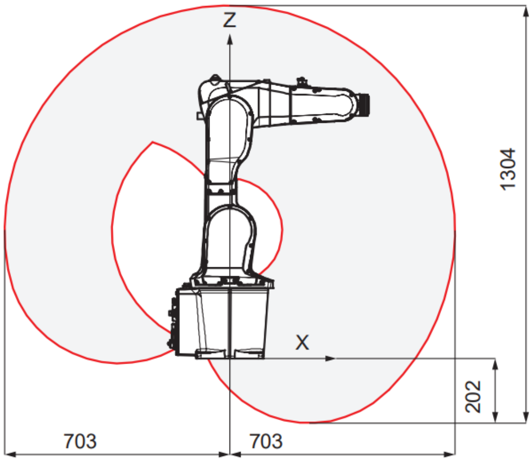

Fig 1. The working range of the IRB 1200 robot

Table 1. *Features of ABB IRB 1200*

Features\name	IRB 1200
Dimensions of the robot	210*210 mm
The weight of the robot	52 kg
Weightlifting capability	5–7 kg
The maximum achievable distance	0.7 m
Maximum operating speed of 1 kg of girth cargo	0.4 s
Number of articulations	6

3. The calculation part of the manipulation robot main components

The important points of our calculations will be the detection of systematic integrity and feasibility of using the chosen system. In the future, we will consider the necessary indicators and accordingly calculate the elements necessary for us for a general understanding of the principles of application of the object according to the indicators [16].

Let's start with the planned schematic display of the robot and the design of the coordinates of the direction of each functional element. As we can see in fig. 2, 6 articulations and their 5 fixed translations were defined. The construction of coordinates and reversible processes will be based on the kinematic dependence of the robot, respectively, which was depicted in fig. 3. Articulations and elongations that were equal were also depicted in the drawings:

$$d_1 = 399,\ a_2 = 350,\ a_3 = 42,\ d_4 = 351,\ d_6 = 82$$
$$q_1(+170/-170),\ q_2(+135/-100),\ q_3(+70/-200),$$
$$q_4(+270/-270),\ q_5(+130/-130),\ q_6(+360/-360)$$

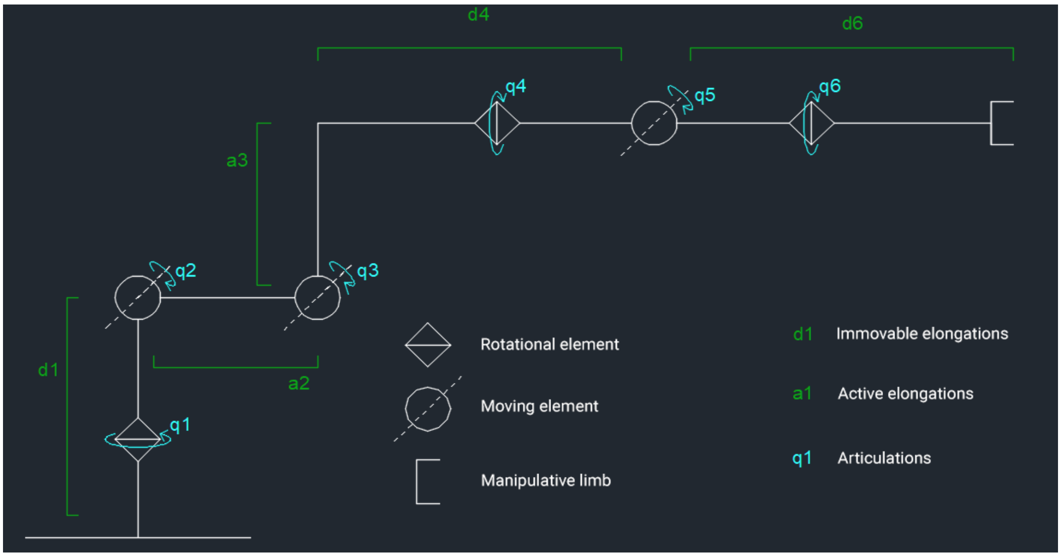

Fig 2. Schematic representation of robot kinematics

Having analyzed the parameters depicted in the design drawings, we can reproduce the table of robot features as one of the options for practice for calculating direct kinematics, considering the Denavit-Hartenberg (DH) parameters of the IRB 1200 robot model. The results were entered into the table. 2 and on figure 3.

In the future, we will determine the indicators of forward and reverse kinematics in accordance with the given table. Adhering to the rules of calculation of elements defined by us. Matrix transformations for direct kinematics can be obtained as follows [17]:

$$ {}^{i-1}_{\ \ i}T = \begin{bmatrix} Cq_i & -Sq_iC\alpha_i & Sq_iC\alpha_i & \alpha_iCq_i \\ Sq_i & Cq_iC\alpha_i & Cq_iS\alpha_i & \alpha_iSq_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Сучасний стан наукових досліджень та технологій в промисловості. 2024. №4 (30)

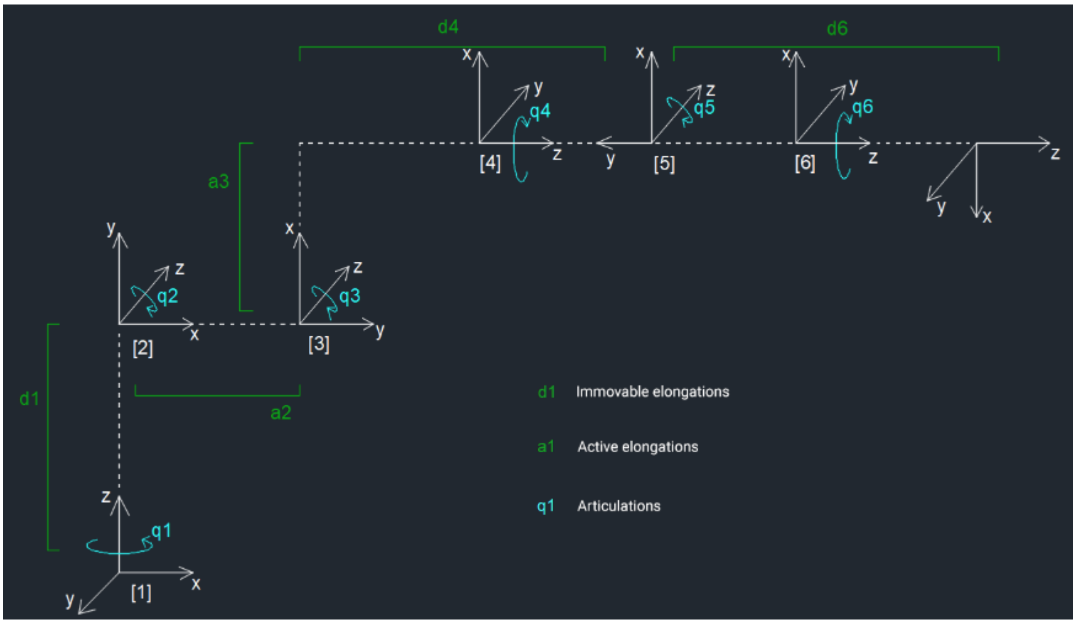

Fig. 3. Schematic representation of robot kinematics

Table 2. *Denavit-Hartenberg parameter of IRB 1200 robot*

	α (degrees)	d (mm)	a (mm)	q (degrees)
1	–90	d_1	0	q_1
2	0	0	a_2	$q_2 - 90$
3	–90	0	a_3	q_3
4	+90	d_4	0	q_4
5	–90	0	0	q_5
6	0	d_6	0	$q_6 - 180$

In the Denavit-Hartenberg (DH) convention for robotic kinematics, C and S are shorthand notations for cosine and sine functions, respectively.

Specifically:

– C represents cos(θ), where θ is the joint angle.

– S represents sin(θ), where θ is the joint angle.

Following the given formula, we derive the transformation matrix for each of the defined articulations. Having received the results for each of the articulations, we calculate the general matrix T, which indicates the location of the system associated with the end of the robot relative to the frame of reference of the robot base, which informs us about the exact position according to direct kinematics. The formula of the general position is as follows:

$$T = {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T\,{}^5_6T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Matrices of transformations (articulations) are presented below:

$$\begin{gathered}
{}^0_1T = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^1_2T = \begin{bmatrix} S_2 & C_2 & 0 & a_2S_2 \\ C_2 & S_2 & 0 & -a_2C_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^2_3T = \begin{bmatrix} C_3 & 0 & -S_3 & a_3C_3 \\ S_3 & 0 & C_3 & a_3S_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^3_4T = \begin{bmatrix} C_4 & 0 & S_4 & 0 \\ S_4 & 0 & -C_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^4_5T = \begin{bmatrix} C_5 & 0 & -S_5 & 0 \\ S_5 & 0 & C_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^5_6T = \begin{bmatrix} -C_6 & S_6 & 0 & 0 \\ -S_6 & -C_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{gathered} \quad (3)$$

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2024. No. 4 (30)

The preceding equations represent the position (px, py, pz) and orientation (n, o, a) values of the end of the robot as a function of the joint coordinates q.

The main equations for the general matrix T calculations are presented below:

$$
\begin{aligned}
n_x &= \left(S_6\left(S_4\left(C_1C_2S_3 + C_1C_3S_2\right) - C_4S_1\right) - C_6\left(C_5\left(S_1S_4 + C_4\left(C_1C_2C_3 + C_1C_3S_2\right)\right) + S_5\left(C_1C_2C_3 - C_1S_2S_3\right)\right)\right)\\
n_y &= \left(S_6\left(S_4\left(C_2S_1S_3 + C_3S_1S_2\right) + C_4C_1\right) - C_6\left(C_5\left(C_4\left(C_2S_1S_3 + C_3S_1S_2\right) - C_1S_4\right) + S_5\left(C_2C_3S_1 - S_1S_2S_3\right)\right)\right)\\
n_z &= -C_6\left(S_5\left(C_2S_3 - C_3S_2\right) - C_4C_5\left(S_2S_3 + C_2C_3\right)\right) - S_4S_6\left(S_2S_3 + C_2C_3\right)\\
o_x &= C_6\left(S_4\left(C_1C_2S_3 + C_1C_3S_2\right) - C_4S_1\right) - S_6\left(C_5\left(S_1S_4 + C_4\left(C_1C_2S_3 + C_1C_3S_2\right)\right) + S_5\left(C_1C_2C_3 - C_1C_2S_3\right)\right)\\
o_y &= C_6\left(S_4\left(C_2S_1S_3 + C_3S_1S_2\right) + C_1C_4\right) - S_6\left(C_5\left(C_4\left(C_2S_1S_3 + C_3S_1S_2\right) - C_1S_4 + S_5\left(C_2C_3S_1 - S_1S_2S_3\right)\right)\right)\\
o_z &= -S_6\left(S_5\left(C_2S_3 - C_3S_2\right) - C_4C_5\left(S_2S_3 + C_2C_3\right)\right) - C_6S_4\left(S_2S_3 + C_2C_3\right)\\
a_x &= C_5\left(C_1C_2C_3 - C_1S_2S_3\right) - S_5\left(S_1S_4 + C_4\left(C_1C_2S_3 + C_1C_3S_2\right)\right)\\
a_y &= C_5\left(C_2C_3S_1 - S_1S_2S_3\right) - S_5\left(C_4\left(C_2S_1S_3 + C_3S_1S_2\right) - C_1S_4\right)\\
a_z &= C_5\left(C_2S_3 - C_3S_2\right) + C_4S_5\left(S_2S_3 + C_2C_3\right)\\
p_x &= C_1a_2S_2 + d_6\left(C_5\left(C_1C_2C_3 - C_1S_2S_3\right) - S_5\left(S_1S_4 + C_4\left(C_1C_2S_3 - C_1C_3S_2\right)\right)\right) + d_4\left(C_1C_2C_3 - C_1S_2S_3\right) + C_1C_2a_3C_3 + C_1S_2a_3C_3\\
p_y &= S_1a_2S_2 + d_6\left(C_5\left(C_2C_3S_1 - S_1S_2S_3\right) - S_5\left(C_4\left(C_2S_1S_3 + C_3S_1S_2\right) - C_1S_4\right)\right) + d_4\left(C_2C_3S_1 - S_1S_2S_3\right) + C_2S_1a_3C_3 + S_1S_2a_3C_3\\
p_z &= a_2C_2 + d_1 - C_2a_3C_3 + d_4\left(C_2S_3 - C_3S_2\right) - S_2a_3C_3 + d_6\left(C_5\left(C_2S_3 - C_3S_2\right) + C_4S_5\left(S_2S_3 + C_2C_3\right)\right)
\end{aligned}
\tag{4}
$$

Next, we will consider inverse kinematics. T_0 adjust the position and orientation of the final mechanism of the robot, to reach the target object, it is most necessary to find the value of inverse kinematics. For a given position of the end effector, the connection angles that move the end effector to the specified position are determined by the inverse kinematic model. In other words, given the given ${}^0_6T$ as sixteen numerical values, the inverse kinematic solution is the corresponding articulations q_1, q_2, q_3, q_4, q_5, q_6. In the future, the kinematic solution is considered purely algebraically from direct kinematic equations:

$$
\begin{aligned}
p_x &= p_x - d_6a_x = C_1C_{23}d_4 + a_3C_1S_{23} + C_1a_2S_2\\
p_y &= p_y - d_6a_y = S_1C_{23}d_4 + a_3S_1S_{23} + S_1a_2S_2\\
\frac{p_y}{p_x} &= \frac{S_1\left(C_{23}d_4 + a_{31}S_{23} + a_2S_2\right)}{C_1\left(C_{23}d_4 + a_{31}S_{23} + a_2S_2\right)}
\end{aligned}
\tag{5}
$$

We will also try to display the exact center of the robot, considering its exact coordinates, which are displayed in the description of the robot. Finding the center of the robot will help us find out the parameters by which we will be able to orient ourselves on how to better place the points of support and movement. In this way, the robot will be balanced and able to support the maximum weight, with which the small robot will be able to move a weight that is competitive enough and receptive enough for small productions or productions with light materials and short reach. Such a robot can be aimed at automation processes with fast production and low economic capacity. Highly recommended for private businesses with a small volume of work for full automation and easier control.

So, to work out the calculations, we will set the following data:

– $d_1 = 290$ mm, $d_4 = 302$ mm, $d_6 = 72$ mm;
– $a_2 = 360$ mm, $a_3 = 70$ mm;
– $q_1 = q_2 = q_3 = q_4 = q_5 = q_6 = 0$.

$$
{}^0_3T^{-1}\begin{bmatrix} n_x & o_x & a_x & p_x\\ n_y & o_y & a_y & p_y\\ n_z & o_z & a_z & p_z\\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^3_6T
$$

$$
{}^0_3T = {}^0_1T\,{}^1_2T\,{}^2_3T = \begin{bmatrix} C_1S_{23} & S_1 & C_1C_{23} & a_3C_1S_{23} + a_2C_1S_2\\ S_1S_{23} & -C_1 & S_1C_{23} & a_3S_1S_{23} + a_2S_1S_2\\ C_{23} & 0 & -S_{23} & a_3C_{23} + a_2C_2 + d_1\\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^0_3T = \begin{bmatrix} {}^0_3R & {}^0_3P\\ 0 & 1 \end{bmatrix}
\tag{6}
$$

$$
{}^0_3T^{-1} = \begin{bmatrix} {}^0_3R^T & -{}^0_3R^T & {}^0_3P\\ 0 & 1 & \end{bmatrix}
$$

$$
{}^0_3T^{-1} = \begin{bmatrix} C_1S_{23} & S_1S_{23} & C_{23} & -a_3 - a_2C_3 - d_1C_{23}\\ S_1 & -C_1 & 0 & 0\\ C_1C_{23} & S_1C_{23} & -S_{23} & a_2S_3 + d_1S_{23}\\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2024. №4 (30) ISSN 2524-2296 (online)

Setting q indicators to 0 was because we estimate the position of the robot in the initial coordinates with zero values, which directly affects the accurate calculation of all positions and the middle of the robot's support. Based on this, we will receive an updated DH table (table 3) with exact coordinates and data that will help us for current calculations.

In the future, we will make calculations following first the calculations of each link in the coordinate system of the robot base, and later the calculation of the final position using the transformation matrices for each link.

As a result of the final coordinates, we will calculate the final coordinates of the position of the robot and its average coordinates. To do this, multiply the matrices step by step:

$$T = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6 \tag{7}$$

Table 3. *DX parameters of the IRB 1200 robot model*

	α (degrees)	d (mm)	a (mm)	q (degrees)
1	–90	290	0	0
2	0	0	360	0–90
3	–90	0	70	0
4	+90	302	0	0
5	–90	0	0	0
6	0	72	0	0–180

Let's calculate matrix multiplication A_2:

$$(A_1 \cdot A_2) : A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 290 \\ 0 & 0 & 0 & 1 \end{bmatrix} A_2 = \begin{bmatrix} 0 & 1 & 0 & 360 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$A_1 \cdot A_2 = \begin{bmatrix} 0 & 1 & 0 & 360 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 290 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

Let's calculate the multiplication of the result by A_3:

$$(A_1 \cdot A_2) \cdot A_3 = \begin{bmatrix} 0 & 0 & 1 & 360 \\ -1 & 0 & 0 & -70 \\ 0 & -1 & 0 & 290 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

Calculate the multiplication of the result by A_4:

$$((A_1 \cdot A_2) \cdot A_3) \cdot A_4 = \begin{bmatrix} 0 & 1 & 0 & 58 \\ -1 & 0 & 0 & -70 \\ 0 & 0 & 1 & 290 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

And calculate the multiplication of the result by A_5:

$$(((A_1 \cdot A_2) \cdot A_3) \cdot A_4) \cdot A_5 = \begin{bmatrix} 0 & 0 & 1 & 58 \\ -1 & 0 & 0 & -70 \\ 0 & -1 & 0 & 290 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

Then by A_6:

$$((((A_1 \cdot A_2) \cdot A_3) \cdot A_4) \cdot A_5) \cdot A_6 = \begin{bmatrix} 0 & 0 & 1 & 130 \\ -1 & 0 & 0 & -70 \\ 0 & -1 & 0 & 290 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

As a result, we considered the issue of forward and inverse kinematics using the parameters of the DX and differential Jacobian matrix. The results were described through a matrix value and expressed as exact coordinates of the position of the robot's limbs and body. The display of the position in the robot zone gives us conclusions about the correctness of the calculations and the fidelity of the construction of the robot. Thus, we can continue to implement our work and analyze its performance.

4. Practical implementation of kinematic model calculated parameters

As described earlier, our goal will be precisely to fit the robot into an already existing virtual production environment. For this, we need to display the standard process of supplying the element – plastic containers to the production area. Suppose that we have a production process for medicines that require separate processing before moving to the production line [18]. Pre-delivery was done by unloading the food machine by worker A, moving the product respectively to two conveyors for pre-treatment, corresponding to the colors of the product and appropriate disinfection with solutions. Worker B created the movement of already processed products on the line.

As we can see, the system works well except that worker B encounters already processed products, thus putting himself and, accordingly, the products at risk, thus it would be much more efficient in this situation to automate the very process of moving the processed products to production line. Based on this, let's assume that this system contains [19]:

– three conveyor belts;
– blocks with disinfection treatment of elements (function separately);
– pink plastic boxes;
– blue plastic boxes.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2024. No. 4 (30)

As a result, we can shape the system in our opinion so that it looks expedient and comprehensively formed. The result is shown in Fig. 4.

Having formed the initial production system, further automation of this process should be broken down into logical points of reproduction. Accordingly, we will receive a plan for implementing a manipulator robot under the appropriate system:

– we form manipulative limbs of the robot, according to the existing plastic containers, program them as an active tool for interaction;

– we install the robot in the production system and adjust the limits of interaction accordingly, attach the limbs;

– we install sensors to control conveyor tracks;

– we install sensors for manipulative limbs;

– we program the movement of parts and the interaction of the manipulator with them;

– we program the sequence of the process, allocate time management and check how the system functions.

First, we will form a limb for the robot, which will accordingly be built based on the previously formed box. The manipulative limb will be due to the minimum number of parts to simplify the work of the robot. For the light weight of the parts, we will make notches in the middle parts for better interaction, reducing friction and better endurance of the boxes by changing the load on the end blocks. The results are shown in fig. 5.

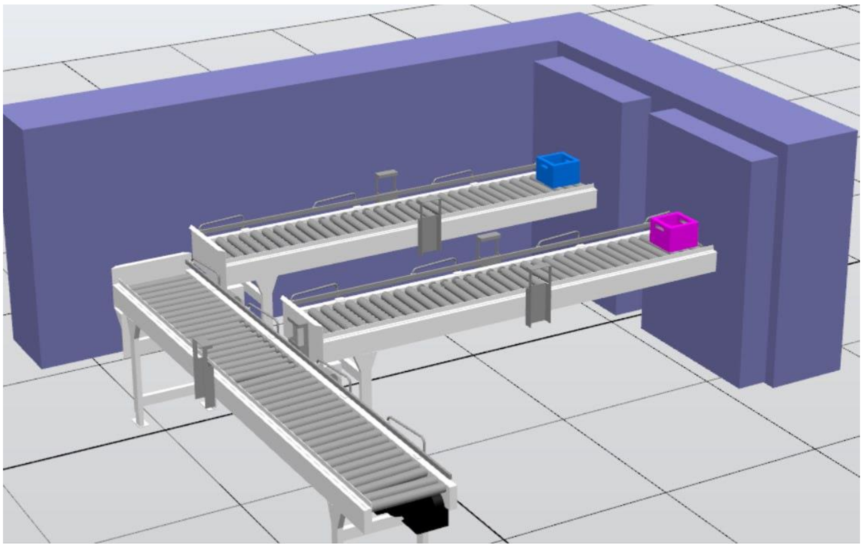

Fig. 4. Initial production system

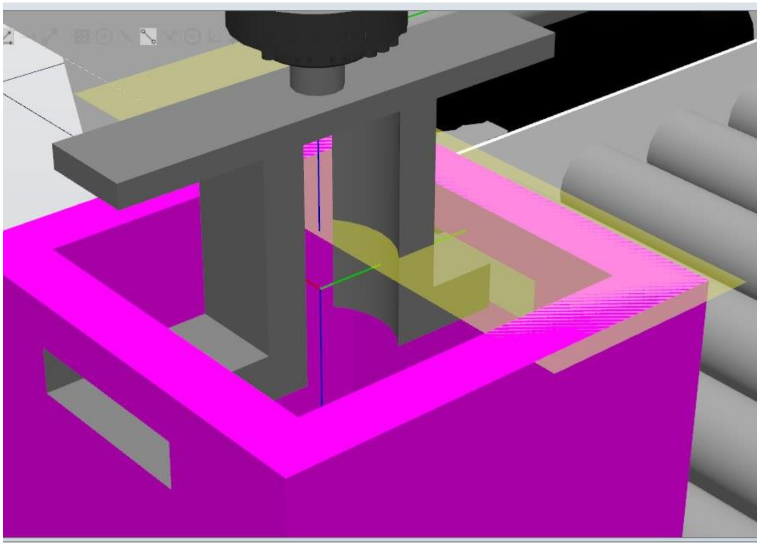

Fig. 5. Design of manipulative limbs

In addition, we will evaluate the size of the box and the robot's manipulative limb for further analysis of the robot's endurance.

The dimensions of the box are equal:

– width: 200 mm;
– height: 150 mm;
– length: 200 mm.

Thus, for the girth of the box, the manipulative limb must adjust to the box itself.

Regarding the estimation of the weight of the box, considering the dimensions and content of the filling – nutritious herbs, the box will not exceed 3 kilograms, since, considering the material of the box – plastic, the weight will be approximately 1.5–2 kilograms. Considering the herbs that will, let's say, fill the box by one third, and add a maximum of 1–1.5 kilograms to the permissible weight, the total weight will not exceed 3 kilograms.

Considering the parameters of the robot, the maximum possible value of lifting materials is 7 kilograms, thus we will be able to calculate the maximum permissible lifting weight by calculating the force [20].

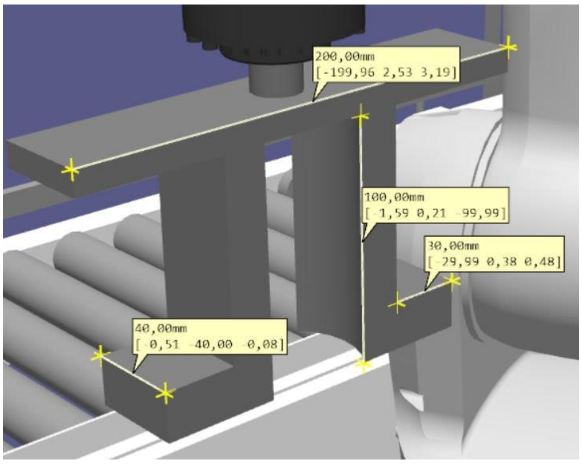

Fig. 6. Calculation details of the manipulated limb

Adjusting the robot IRB 1200_5_90_STD_03 and the controller IRB1200_5_90 into the already existing process, we put it on a stand, setting a larger radius of interaction with conveyors (Fig. 7). Which will be equal as a result:

– width: 500 mm;
– length: 500 mm;
– height: 600 mm.

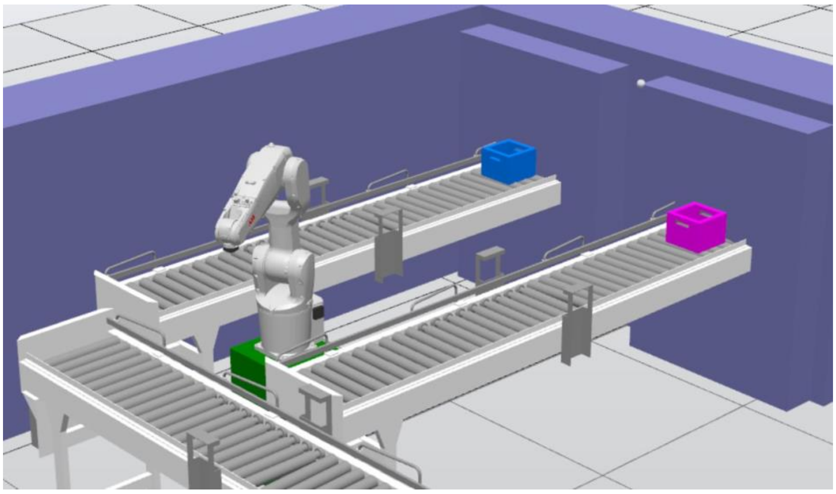

Fig. 7. Placing the robot in the system

Such parameters for the stand were set according to the height of the conveyor belt itself, which reaches 750 mm. Considering the statistical stand of the robot, the final height of the robot is compensated and as a result is equal to the optimal placement of the robot in space. We attach the manipulative limb, respectively, and set the interaction parameters, obtaining a finished model of the robot (Fig. 8).

Next, we will specify step-by-step methods of operating conveyors. In this work, I will use one of the new functional solutions of Robot Studio – SmartComponent, which allows us to adjust functional objects quickly and efficiently. SmartComponent is one of the key functions in the RobotStudio software, which is used to simulate, program, and optimize the operation of ABB industrial robots. This feature allows users to create modular, multi-functional components that can be easily integrated into virtual environments to improve simulation accuracy and efficiency.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Innovative technologies and scientific solutions for industries. 2024. No. 4 (30)

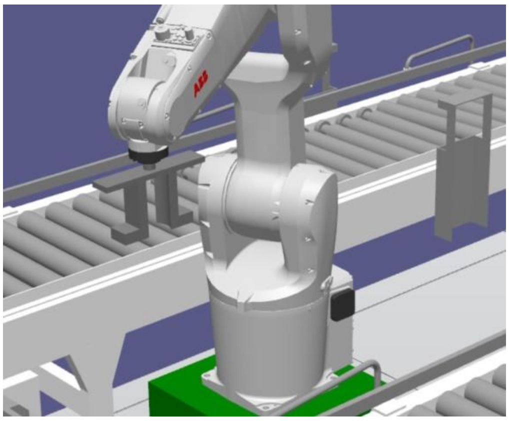

Fig. 8. The final appearance of the robot model

The overall conveyor system will consist of a sensor [21, 22], a linear motor, and an object rendering element. But the conveyor of the final line will consist only of a mover and a sensor that will move exactly the object placed on the conveyor (Fig. 9–11).

Next, we set the input and output signals for the reproduction of the process, form a block diagram of the interaction of the robot with objects (Fig. 12), form the code for successive movements and adjust the playback time.

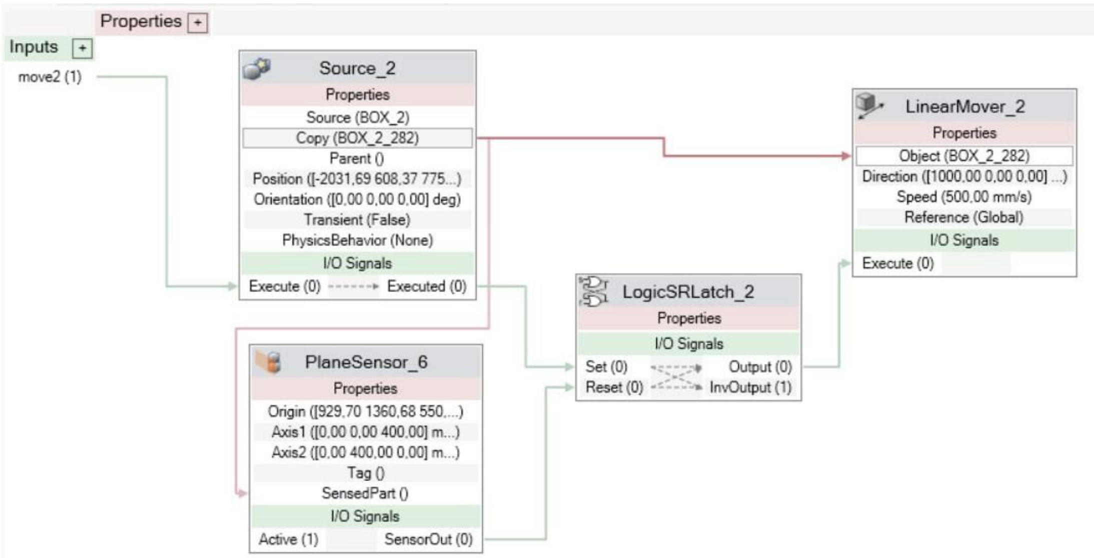

Fig. 9. Conveyor for processing pink boxes

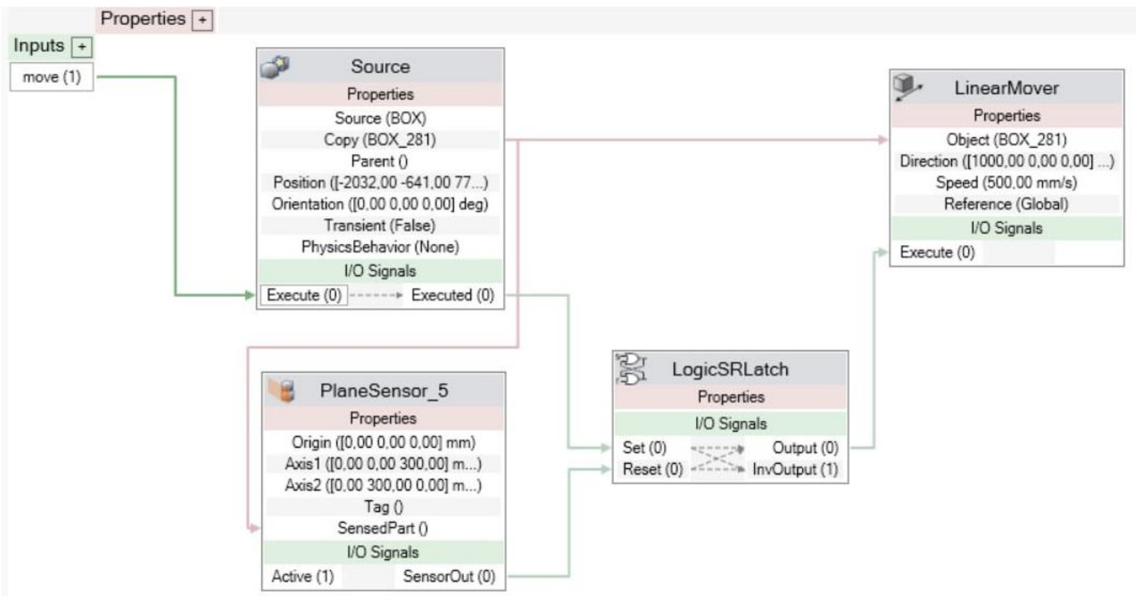

Fig. 10. Conveyor for processing blue boxes

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2024. №4 (30) ISSN 2524-2296 (online)

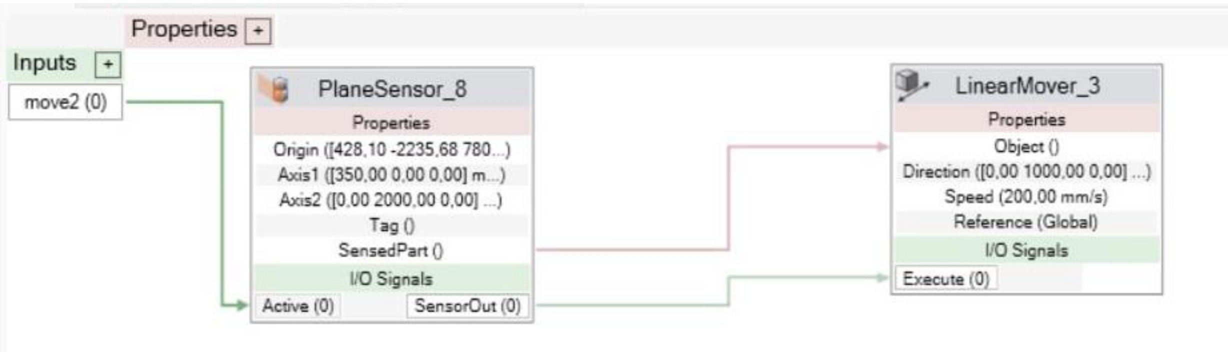

Fig. 11. Conveyor for the final movement of boxes

Fig. 12. Scheme of robot interaction with objects

Input signals are used to receive information from various devices and sensors. They provide feedback, which is necessary for making decisions and adjusting robot actions, as well as output signals. Thus, it is important for us to specify input and output signals for sensors and controlled movement routes. To facilitate the perception of the presented signals, it is important to clarify that most of the presented signals are formed due to automated sensor and circuit reconstruction links.

Circuits, in turn, only have an input, which is used to regulate the start and end of work, without executing the complete system, you can always separate individual elements of these circuits, and try to run them independently, if the system requires it.

Thus, when reproducing the assembled system, everything functions according to the idea of operation, the movement is carried out by common detectors with the control of switching the time of delivery of objects for processing by the robot and the time of movement of the conveyor final belt, thus, the system is not reloaded and supplies objects evenly for further processing of goods (Fig. 13).

The rearrangement of objects takes place due to automated processes, but it is important to emphasize that the passage of boxes onto the conveyors itself occurs through mechanical interventions, these processes can be synchronized, but during the processing of the project it was found that the best method would be to have a controlled element that would help accordingly in working out the system [23, 24].

As we can see, the robot performs successive steps and adapts well to the production process itself. The integration of a manipulator robot into a production system requires careful planning and the sequential execution of a few steps.

From the development of manipulative limbs to the programming of processes and the installation of sensors, each step is critical to ensure the efficient and safe operation of the system. By following all these steps, you can achieve high accuracy and productivity of an automated production system.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) *Innovative technologies and scientific solutions for industries. 2024. No. 4 (30)*

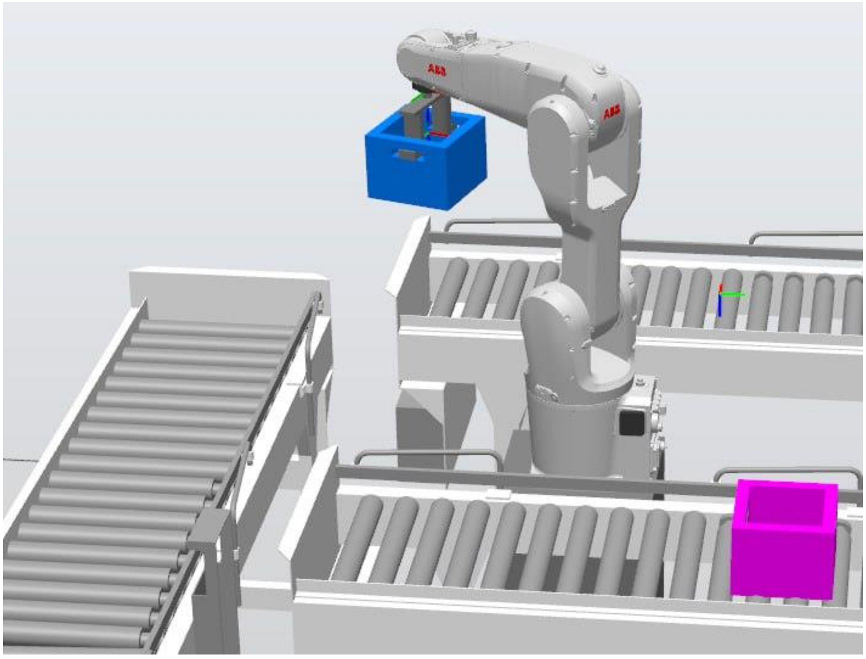

Fig. 13. Examples of robot work

.

Conclusions

This work can be developed using physical development capabilities and successive implementation into a functional system using a physical model control panel with a reproduced model of manipulative limbs. Also, including the control panel in the system, it will be possible to adjust the position of the robot at any point in the automated production.

The manipulative element created can be transferred to 3D printing or additive manufacturing to translate the physical model into production. So, in general, the project has room for development and future implementation, given the minimal economic burden and great opportunities to save money in a long period of time, can be used by anyone and improved, adapted to the respective projects, using the step-by-step tasks of this work.

The developed model and software module for the adaptive execution of the production process as part of the virtual model of the ABB IRB 1200 robot. Thanks to the introduction of the adaptive manipulator robot, it was possible to increase the efficiency of the production process, reduce downtime and improve product quality. The optimization of trajectories and collision management allowed to reduce the maintenance and operation costs of the robot, which led to an overall reduction in production costs, which will also be reflected in long-term use. The use of virtual tools and simulation helped to identify and eliminate potential hazards during the planning and production process, which significantly improved the safety of the system. The developed software module allows the work to quickly adapt to changes in the production process, which provides high flexibility and the ability to quickly adjust to new tasks.

These results confirm the expediency of introducing adaptive robotic systems into production processes and open new perspectives for the further development and improvement of automation technologies.

References

1. Bendeberya, M.O. (2023), "Development of an algorithmic and functional model of a robot manipulator based on ABB Robot Studio" ["Rozrobka alhorytmichno-funktsionalʹnoyi modeli robota manipulyatora na bazi ABB Robot Studio"]. *Automation and development of electronic devices*, Vol 2. P. 65–67. available at: https://tapr.nure.ua/wp-content/uploads/2023/11/zbirnik-aded2023_2.pdf
2. Bendeberya, M.O. (2024), "Development of an algorithmic and functional model of a robot manipulator based on ABB Robot Studio" ["Rozrobka alhorytmichno-funktsionalʹnoyi modeli robota manipulyatora na bazi ABB Robot Studio"]. *Automation and development of electronic devices*, Vol 1. P. 63–65. available at: https://drive.google.com/file/d/1OZ5_Dtlf_s--bk5_3-NAP0QUx9mAjCrf/view

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

3. Qiao, Lijun; Xiao, Luo; Qingsheng, Luo; Minghao, Li; and Jianfeng, Jiang (2021), "Optimizing Kinematic Modeling and Self-Collision Detection of a Mobile Manipulator Robot by Considering the Actual Physical Structure" *Applied Sciences* 11, no. 22: 10591 p. DOI: https://doi.org/10.3390/app112210591
4. Singh, A.; Singla, A. (2017), "Kinematic Modeling of Robotic Manipulators". *Proc. Natl. Acad. Sci., India, Sect. A Phys. Sci.* 87, P. 303–319. DOI: https://doi.org/10.1007/s40010-016-0285-x
5. Kirćanski, M. (1989), Manipulator Kinematic Model. In: Introduction to Robotics. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-642-82997-0_2
6. Ju, R. Zhang, D., Xu, J., Yuan, H., Miao, Z., Zhou, M., and Cao, Z. (2022), "Design, Modeling, and Kinematics Analysis of a Modular Cable-Driven Manipulator." ASME. *J. Mechanisms Robotics*. December 2022; 14(6): 060903. DOI: https://doi.org/10.1115/1.4054206
7. Ceccarelli, Marco; Erika, Ottaviano. (2008), "Kinematic Design of Manipulators". *Robot Manipulators. InTech*. DOI:10.5772/6198
8. Haugaløkken, B. O. A.; Nissen, O.; Amundsen, H. B.; FØre, M.; Kelasidi, E. (2021), "Modelling and control of a 6 DOF robot manipulator for underwater applications – aquaculture related case studies," *OCEANS 2021: San Diego – Porto*, San Diego, CA, USA, P. 1–10, DOI: 10.23919/OCEANS44145.2021.9706100
9. Dong, Yang; Jianzhong, Ding; Chunjie; Wang; Xueao, Liu (2021), "Kinematics Analysis and Optimization of a 3-DOF Planar Tensegrity Manipulator under Workspace Constraint" *Machines* 9, no. 11: 256 p. DOI: https://doi.org/10.3390/machines9110256
10. Jie, Zhao; Cuncun, Wu; Guilin, Yang; Chin-Yin, Chen; Silu, Chen; Ciyuan, Xiong; Chi, Zhang (2022), Kinematics analysis and workspace optimization for a 4-DOF 3T1R parallel manipulator, *Mechanism and Machine Theory*, Volume 167. DOI: https://doi.org/10.1016/j.mechmachtheory.2021.104484
11. Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2020), Inverse Kinematics, Robot modeling and control second edition. P. 141–162. Chichester, West Sussex, UK: John Wiley & Sons, Ltd.
12. "ABB Robotics". available at: https://new.abb.com/products/robotics (last accessed 10.07.2024).
13. Raina, D., Saha, S. K. (2021), "AI-Based Modeling and Control of Robotic Systems: A Brief Tutorial," 3rd International Conference on Robotics and Computer Vision (ICRCV), Beijing, China, P. 45–51, DOI: 10.1109/ICRCV52986.2021.9546974
14. Ermolov, Ivan (2020), "Industrial Robotics Review". *Robotics: Industry 4.0 Issues & New Intelligent Control Paradigms*, P. 195–204, DOI:10.1007/978-3-030-37841-716
15. Buri, Zsolt (2024), "Present and Future of Industrial Robotisation" *International Journal of Engineering and Management Sciences*. P. 1–12. DOI: https://doi.org/10.21791/IJEMS.2024.010
16. Tantawi, K., Sokolov, A., Tantawi, O. (2019), "Advances in Industrial Robotics: From Industry 3.0 Automation to Industry 4.0 Collaboration". *4th Technology Innovation Management and Engineering Science International Conference (TIMES-iCON)*, P. 1–4. DOI: 10.1109/TIMES-iCON47539.2019.9024658
17. Soori, M., Dastres, R., Arezoo, B., Karimi Ghaleh Jough F. (2023), "Intelligent Robotic Systems in Industry 4.0, A Review". *Journal of Advanced Manufacturing Science and Technology*. DOI: 10.51393/j.jamst.2024007
18. Sethuraman, S., Vikram, S., Surya, R., Singh, B. (2024), "Applications of Cloud Computing in Industrial Robotics", *Shaping the Future of Automation With Cloud-Enhanced Robotics*, P. 181–204. DOI:10.4018/979-8-3693-1914-7.ch010
19. Alexandra, A. (2021), "Perspectives of virtual commissioning using ABB RobotStudio and Simatic robot integrator environments: a review". *Proceedings in Manufacturing Systems*, Vol. 16, Issue 3, P. 117–123. available at: https://www.researchgate.net/publication/363454592_Perspectives_of_virtual_commissioning_using_abb_robotstudio_and_simatic_robot_integrator_environments_a_review
20. Corke, P. (2019), "Manipulator Velocity: Robotics, Vision and Control", *Springer, Cham*. 229 p. available at: https://link.springer.com/book/10.1007/978-3-031-07262-8
21. Leonard, Kevin & Seuferling, Tess. (2021), "Electrochemical Instrumentation", *Encyclopedia of Electrochemistry*, P. 1–33. DOI:10.1002/9783527610426.bard030104
22. Nevliudov, I., Andrusevich, A., Yevseev, V., Novoselov, S., Demska, N. (2022), "Design of mobile manipulation robots" ["Proektuvannya mobilʹnykh manipulyatsiynykh robotiv"]: Monograph, *NURE publishing*, 427 p. available at: https://catalogue.nure.ua/document=265298
23. Nevliudov, I., Andrusevich, A., Yevseev, V. (2020), "Designing mobile robots based on single-board computers (Raspberry Pi and Python 3.6)" ["Proektuvannya mobilʹnykh robotiv na bazi odnoplatnykh kompʺyuteriv (Raspberry Pi i movy Python 3.6)"], *NURE publishing*, 257 p. available at: https://catalogue.nure.ua/document=265300
24. Tsymbal, O., Brinnikov, A. (2019), "Robot adaptation systems and OpenCV technology" ["Systemy adaptatsiyi robotiv i tekhnolohiya OpenCV"], *NURE publishing*, 144 p.: ISBN 978-966-659-268-5. available at: https://catalogue.nure.ua/document=235947

Надійшла (Received) 11.11.2024

Відомості про авторів / About the Authors

Бронніков Артем Ігорович – кандидат технічних наук, Харківський національний університет радіоелектроніки, доцент кафедри комп’ютерно-інтегрованих технологій, автоматизації та робототехніки, Харків, Україна; e-mail: artem.bronnikov@nure.ua; ORCID ID: 0000-0003-3096-7653

Бендеберя Марія Олександрівна – Харківський національний університет радіоелектроніки, студентка, кафедра комп’ютерно-інтегрованих технологій, автоматизації та робототехніки, Харків, Україна; e-mail: mariia.bendeberia@nure.ua; ORCID ID: 0009-0004-7528-9264

Bronnikov Artem – PhD (Engineering Sciences), Kharkiv National University of Radio Electronics, Associate Professor at the Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv, Ukraine.

Bendeberia Mariia – Kharkiv National University of Radio Electronics, Student, Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv, Ukraine.

РОЗРОБЛЕННЯ КІНЕМАТИЧНОЇ МОДЕЛІ МАНІПУЛЯТОРА НА БАЗІ *ABB ROBOT STUDIO*

Предметом дослідження в статті є застосування технології та алгоритмів керування, побудови траєкторії рухів і взаємодії з виробничим середовищем на базі *Abb Robot Studio*. **Мета роботи** – симуляція кінематичної моделі маніпулятора, що дає змогу ефективно виконувати дії в реальному часі, застосовуючи проведені математичні розрахунки та програмне забезпечення *Abb Robot Studio*. У статті необхідно виконати такі **завдання**: проаналізувати сучасний стан інтеграції маніпуляторів у виробничі процеси; обрати й обґрунтувати платформу розроблення та модель робота; запропонувати кінематичну модель та спроєктувати й написати програмне забезпечення для керування маніпуляторами на основі проведених математичних розрахунків. Використані **методи**: математичний аналіз і теорія автоматичного керування для моделювання роботи промислового робота; моделювання – для перевірки роботи моделі промислового робота в середовищі *Abb Robot Studio*. Досягнуто таких **результатів**: проаналізовано сучасний стан інтеграції маніпуляторів у виробничі процеси, зокрема в епоху *Industry* 4.0; обрано та обґрунтовано модель маніпуляційного робота, що задовольняє проведені математичні розрахунки; запропоновано кінематичну модель та проведено її проєктування; розроблено програмне забезпечення для керування маніпуляторами на основі проведених математичних розрахунків. **Висновки**: розроблена модель і програмний модуль для адаптивного виконання виробничого процесу в складі віртуальної моделі робота *ABB IRB* 1200, що дає змогу підвищити ефективність виробничого процесу, зменшити час простою та покращити якість продукції. Оптимізація траєкторій та управління колізіями уможливили зниження витрат на обслуговування та експлуатацію робота, що привело до загального зниження виробничих витрат, які також матимуть місце за умови тривалого використання. Розроблений програмний модуль дає змогу роботу швидко адаптуватися до змін у виробничому процесі, що забезпечує високу гнучкість і можливість оперативного переналаштування під нові завдання.

Keywords: *Industry* 4.0; *Abb Robot Studio*; маніпулятор; робот; виробництво; кінематична модель; математичні розрахунки.

Бібліографічні описи / Bibliographic descriptions

Бронніков А. І., Бендеберя М. О. Розроблення кінематичної моделі маніпулятора на базі *Abb Robot Studio*. *Сучасний стан наукових досліджень та технологій в промисловості*. 2024. № 4 (30). С. 5–18. DOI: https://doi.org/10.30837/2522-9818.2024.4.005

Bronnikov, A., Bendeberia, M. (2024), "Development of a manipulator kinematic model using Abb Robot Studio", *Innovative Technologies and Scientific Solutions for Industries*, No. 4 (30), P. 5–18. DOI: https://doi.org/10.30837/2522-9818.2024.4.005