ТЕХНИЧЕСКИЕ НАУКИ
И ТЕХНОЛОГИИ

ИНТЕЛЛЕКТУАЛДЫ РОБОТОТЕХНИКАЛЫҚ ЖҮЙЕЛЕР
ИНТЕЛЛЕКТУАЛЬНЫЕ РОБОТОТЕХНИЧЕСКИЕ СИСТЕМЫ
INTELLIGENT ROBOTIC SYSTEMS

DOI 10.51885/1561-4212_2022_1_39
МРНТИ 28.23.27; 28.23.15

A.T. Kadyroldina[1], A.Zh. Orazova[1], A.L. Krasavin[1], I.G. Kazantsev[2], I.A. Dyomina[1], D.L. Alontseva[1]
[1]D.Serikbayev East Kazakhstan Technical University
[2]Institute of Computational Mathematics and Mathematical Geophysics of the Siberian Branch of the Russian Academy of Sciences
*E-mail: akadyroldina@gmail.com**
E-mail: arailym-vko@mail.ru
E-mail: alexanderkrasavin@mail.ru
E-mail: irdyomina@mail.ru
E-mail: dalontseva@mail.ru
E-mail: kig@ooi.sscc.ru

DEVELOPMENT OF NEW CONTROL ALGORITHMS FOR A ROBOTIC ARM EQUIPPED WITH A 3D SCANNING OR MACHINE VISION SYSTEM

3D СКАНЕРЛЕУ НЕМЕСЕ МАШИНАЛЫҚ КӨРУ ЖҮЙЕСІМЕН ЖАБДЫҚТАЛҒАН РОБОТ-МАНИПУЛЯТОРҒА АРНАЛҒАН ЖАҢА БАСҚАРУ АЛГОРИТМДЕРІН ӘЗІРЛЕУ

РАЗРАБОТКА НОВЫХ АЛГОРИТМОВ УПРАВЛЕНИЯ РОБОТА-МАНИПУЛЯТОРА, ОСНАЩЕННОЙ СИСТЕМОЙ 3D СКАНИРОВАНИЯ ИЛИ МАШИННОГО ЗРЕНИЯ

Abstract. *Manipulator robots perform many complex actions and solve important tasks in the industry (welding, cutting, spaying, sorting, etc.). This article discusses the geometric problems of the kinematics of the robot, in order to ensure the smoothness of movement without instantaneous stops at the corners of the trajectory. The problem of equipping the robot with additional 3D scanning or machine vision system to control such situations is considered. It is proposed to use a method for recognizing the dominant points of the trajectory, in particular, corner points, based on image processing with specialized masks.*

Keywords: *robotic arm, smooth trajectories, corner points.*

Аңдатпа. *Робот-манипуляторлар көптеген күрделі іс-қимылдарды орындайды және өнеркәсіптегі маңызды міндеттерді шешеді (дәнекерлеу, кесу, бүрку, сұрыптау және т.б.). Бұл мақалада траекторияның бұрыштарында лезде тоқтаусыз қозғалыстың тегістігін қамтамасыз ету үшін робот кинематикасының геометриялық мәселелері қарастырылады. Мұндай жағдайларды бақылау үшін роботты қосымша 3D сканерлеу немесе машиналық көру жүйелерімен жабдықтау мәселелері қарастырылады. Суреттерді мамандандырылған маскалармен өңдеуге негізделген траекторияның тірек нүктелерін, атап айтқанда бұрыштық нүктелерді тану әдісін қолдану ұсынылады.*

Түйін сөздер: *робот-манипулятор, тегіс траекториялар, бұрыштық нүктелер.*

Аннотация. *Роботы-манипуляторы выполняют множество сложных действий и решают важные задачи в промышленности (сварка, резка, напыление, сортировка и т.д.). В данной статье* *рассматриваются геометрические проблемы кинематики робота, с целью обеспечить гладкость движения без мгновенных остановок в углах траектории. Рассматриваются проблемы оснащения робота дополнительно системами 3D сканирования или машинного зрения для контроля таких ситуаций. Предлагается использование метода распознавания доминирующих точек траектории, в частности, угловых точек, на основе обработки изображений специализированными масками.*

Ключевые слова: *робот-манипулятор, гладкие траектории, угловые точки.*

Introduction. The robot manipulator has become an integral part of modern industrial automation. Currently, the use of robotic manipulators in industry is steadily expanding, robots are used for coating, loading and packaging operations, on assembly lines of automotive and machine-building enterprises, etc., since their characteristics allow the technological process to be carried out with precision accuracy and high productivity. It is also important that automation of production with the help of robots allows for long-term technological processes in difficult or harmful for human conditions.

Robot manipulators are controlled by programmable controllers. Most industrial controllers use a general control principle - a linear proportional control algorithm for each link, where the spatial position of the working tool is corrected, and the feedback signal is the position of the tool [1]-[2]. This method is suitable for low speeds and requires movement along a given trajectory in time. However, modern production requires more flexible process control in order to increase the speed of passing a given trajectory and set the trajectory without preliminary calculations for each individual link of the robot. A large number of studies all over the world are currently devoted to the solution of this problem, such as papers [3]-[11], however, the solution is not unambiguous and varies depending on the control of specific processes.

Thus, the development of new control methods for a multi-link robot manipulator is in the focus of world scientific research, representing significant scientific and practical interest in relation to specific technological processes.

Works [12]-[16] are devoted to the application of new algorithms for controlling a robot manipulator performing microplasma spraying. In these works, the robot moves according to a digital 3D model of the covered part, while a 3D model of an industrial product is obtained by 3D scanning it by the same robot manipulator.

The purpose of this study was to carry out a theoretical justification for the automatic trajectory planning of the working tool of the robot performing plasma spraying of coatings, to implement the generation of the motion program of the robot manipulator based on 3D scanning data and to consider the issues of equipping the robot with an additional independent machine vision system using the method of recognition of dominant points of the trajectory, in particular, corner points.

Theory and methods of research. As described in authors previous papers [12]-[16], a new scheme of robotic 3D scanning has been developed and implemented. According this scheme, measurements in the nodes of the scanning grid make once, and here is a justification for why single measurements at the nodes of the selected scanning grid are preferable.

Firstly, the speed of triangulation distance sensors is so high that it allows scanning from a moving sensor, without stopping at the nodes of the scanning grid. The controller of the robot manipulator allows reading the current spatial position of the working tool on request, and the applied triangulation distance sensors are equipped with a network interface that allows measurement on request. During the scanning process, a pair of requests are sequentially executed from the control computer - a request to take measurements to the distance sensor and a request to transfer the coordinates of the working tool to the controller of the robot manipulator. The time interval between these two requests is so small that with a reasonable choice of the sensor movement speed, the distance traveled by the sensor during this time interval can be neglected. Such a scanning scheme allows you to significantly increase the scanning performance (in comparison with the scheme in which the manipulator stops at the nodes of the scanning grid) and, moreover, to carry out a large number of measurements necessary for accurate restoration of the surface from the noisy data of the triangulation distance sensor.

Secondly, with an equal total number of measurements, a scheme with single measurements at multiple points distributed over the scanned surface has an advantage in the accuracy of surface reconstruction over a scheme with multiple measurements at more sparse scanning points. When using this scanning scheme, the sensor trajectory is a polyline lying in a horizontal plane. The trajectory consists of U-shaped segments covering the scanning area. In this case, the working sections of the trajectory (segments of a broken line, when moving along which the distance sensor is interrogated) are segments of parallel lines located at equal distances from each other. The scan data is a point cloud. Strictly speaking, this point cloud is an unordered set of triples of Cartesian coordinates of points on the scanned surface. With the chosen scanning scheme, the first pair of coordinates (the position of the sensor on the plane of the sensor trajectory) is measured with a high degree of accuracy, while the third coordinate (the distance from the plane of the sensor trajectory to the point of the scanned surface) can be considered as a random variable, the mathematical expectation of which corresponds to the actual value of this coordinate. Thus, an attempt to restore the surface using direct 2D interpolation methods based on scanning data will lead to unacceptably large errors. In order to overcome these difficulties, a data processing algorithm using regression analysis methods to construct a local parametrizable surface model has been developed [14]. The algorithm [14] includes a surface segmentation procedure, during which local models of overlapping surface segments are combined into regions having a homogeneous geometric structure (described by a general analytical parametrizable model). Such a segmentation procedure is necessary for the further formation of the manipulator trajectory for two reasons: firstly, the methods to form the manipulator trajectory use an analytical surface model as input data; secondly, the use of segmentation methods allows to partially solve the problems of surface reconstruction which shape cannot be described by a smooth function of two variables.

When designing the system of automatic generation of the program of an industrial robot manipulator for applying microplasma coatings to a surface of arbitrary shape, a number of assumptions and limitations concerning both the coating process model and the technological parameters of the process has been taking into consideration. Since the working tool of the manipulator is a movable plasma source forming a plasma jet with a stream of coating particles or a plasma jet of ionized particles for plasma cutting, it was assumed that in any case the plasma jet can be modeled by a flow of particles having the shape of a cone. It was also assumed that the distribution of the particle flux through the plane perpendicular to the axis of the sputtering cone is radially symmetric. These assumptions are confirmed by an experimental study by Borisov Yu.S., et al [17], which provides data on the metallization spot (stationary case) and on the shape of the formed coating. Thus, the data [17] confirmed the choice of a model with a radially symmetric distribution of particles with respect to the cone axis.

There are two mandatory requirements for the trajectory of the manipulator's working tool. First, at any given time, the axis of the spraying cone must be perpendicular to the treated surface. When this requirement is met, the trajectory of the working tool in space will correspond to a curve on the treated surface, which is the geometric location of the points of intersection of the axis of the spraying cone with the surface. In the future, we will call this curve a trace of the trajectory of the manipulator's working tool. The second requirement for the trajectory of the manipulator's working tool was the constancy of the distance from the working tool to the surface throughout the trajectory.

It is also assumed that the treated surface can be described as a piecewise-smooth function, and in areas whose geometric dimensions are of the order of the geometric dimensions of the deposition spot, the surface can be approximated by a plane with a sufficient degree of accuracy.

The generated trajectory of the manipulator's working tool has a U-shaped shape, while the trajectory trace on the surface consists of working segments representing equidistant curves on the surface and segments, during the passage of which no spraying is carried out, but the manipulator maneuver is carried out with the aim of reaching an adjacent working segment of the trajectory. It is assumed that the coating area is bounded by a closed curve on the surface.

Authors apply a trajectory generation scheme in which the working segments of the trajectory are constructed as equidistant copies of one selected working fragment of the trajectory. Summarizing the experience of practical application of methods of forming the trajectory of a working tool of this class, described in [18], [19], it can be argued that the choice of the starting trajectory has a strong influence on both the quality of the coating and the productivity of the processing process.

Speaking about the influence of the choice of the starting trajectory, we mean, first of all, the uniformity of the thickness of the applied coating (the meaning of this statement will be revealed later when substantiating the method of forming the starting trajectory that we use).

Most of the existing trajectory formation algorithms [18]-[23] use the secant plane method to form the starting fragment of the trajectory, in which the line of intersection of the surface with a certain plane is taken as the starting fragment of the trajectory. To form the starting fragment of the trajectory, we use a specially selected geodesic line on the surface. Here are the necessary definitions and justifications for choosing a starting segment of this type.

For curves on surfaces, the concepts of normal and geodesic curvature are separated. Below we give definitions of the concepts of normal and geodesic curvature of a curve on a surface, based on the concept of the curvature vector of the curve. Let $\vec{\tau}(s)$ be a function describing the dependence of the tangent to some smooth curve of a vector of unit length on the natural parameter s. Then the curvature vector of the curve, $\vec{k}$ is determined by the formula (1)

$$\vec{k}(s_0) = \left(\frac{d\vec{\tau}}{ds}\right)|_{s=s_0} \tag{1}$$

Now consider a smooth curve γ on the surface S passing through the point $X = (x, v)$ lying on S. Let $\vec{r} = \vec{r}(s) = r(u(s), v(s))$ be some natural parametrization of the curve γ . At the point X we find the unit vector $\vec{\tau}$ tangent to γ. The analytical expression for $\vec{\tau}$ is given by equation (2)

$$\vec{\tau} = \frac{d\vec{r}}{ds} \tag{2}$$

Denote the normal to the surface S at point X as n, and define the vector $\vec{b} = \vec{n} \times \vec{\tau}$. It is not difficult to show that the triple of vectors $\vec{\tau}$, $\vec{n}$, $\vec{b}$ is linearly independent, which makes it ibpossle to represent the curvature vector of the surface k as a decomposition over the vectors of the basis $\vec{\tau}, \vec{n}, \vec{b}$ (3)

$$\vec{k} = \alpha \cdot \vec{\tau} + \beta \cdot \vec{n} + \delta \cdot \vec{b} \tag{3}$$

It is proved (see for example [24]) that $\alpha = \langle \vec{k}, \vec{\tau} \rangle = 0$, and the coefficients β, δ are determined by equations (4) and (5) and have special names.

$$\beta = k_n = (\vec{k}, \vec{n}) \tag{4}$$

where k_n is called the normal curvature of the surface.

$$\delta = k_g = (\vec{k}, \vec{b}) \qquad (5)$$

where k_g is called the geodesic curvature of the surface.

Obviously, the normal curvature of any curve on the plane is zero at any point of the curve. The term «geodesic line on a surface» is used to denote a curve on a surface whose geodesic curvature is zero at any point. Thus, geodesic lines on the plane will be straight lines. In addition, in the general case, the geodesic line connecting two points on the surface is the curve of the smallest length, of all the curves on the surface passing through these two given points. Thus, geodesic curvature is a measure of curvature in the internal geometry of the surface, and geodesic lines play the same role in the internal geometry of the surface as straight lines in geometry on the plane. Let the curve on the surface be a trace of the spray nozzle. Figure 1 shows three possible cases of the influence of the geodesic curvature of the trace (GCT) of the trajectory on the uniformity of the coating applied.

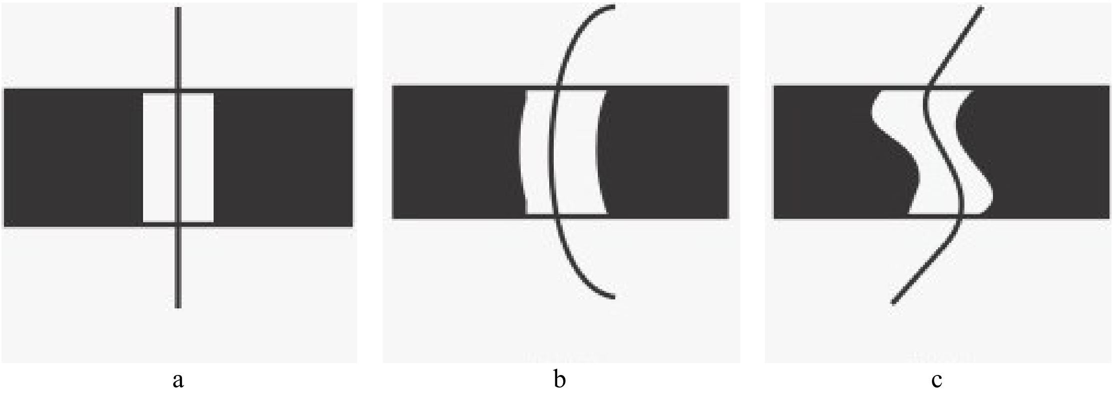

Figure 1. The influence of the geodesic curvature of the trace (GCT) on the uniformity of the coating applied a – GCT = 0, uniform distribution of the coating on both sides of the trace; b – GCT = const, the thickness of the coating increased in the direction of curve bending; c – GCT variable, uneven distribution of the coating on both sides of the trace [18]

It follows from Fig. 1 that in the case (a) when the geodesic curvature of the trace (GCT) is zero, the sprayed material will be distributed evenly on both sides of the trace trajectory. Case (b) corresponds to the constant geodesic curvature of the trajectory. It is obvious that most of the sprayed material will accumulate in the direction of the curve bend. It is also clear that in the case of a variable geodesic curvature of the trace, case (c), the coating thickness will be distributed unevenly on both sides of the trace. This visual examination suggests that the optimal choice of the starting segment of the trajectory will be one of the geodesic lines on the surface. In addition, this consideration shows that it is desirable to choose the starting segment in such a way that the trajectory segments obtained by shifting the starting segment have as little geodesic curvature as possible.

During the development of an automatic trajectory planning system and the generation of a robot manipulator movement program based on the formed three-dimensional model of the product, three ways of generating the starting fragment have been tested. These three ways of generating the starting fragment included two automatic and one semi-automatic. The first automatic method was based on the minimum height criterion proposed in [25], the second, original method was based on sorting the directions of the starting segment and evaluating the quality of the generated trajectories according to an integral criterion, the third method was semi-automatic, in which the operator sets the point and direction of the starting geodesic curve in it.

Let's explain the task of automatically generating a robot manipulator program along a given trajectory of the working tool. The mathematical formulation of the problem will be given as a problem of optimal approximation of a spatial curve by a sequence of geometric primitives with remarks on the software implementation of algorithms for automatic generation of a robot manipulator program.

The result of the procedure for generating the trajectory of the working tool of the robot manipulator is a sequence of points in three-dimensional space, and we can assume that these points lie on a piecewise smooth spatial curve. This sequence of points will be considered as a sequence of vertices of a polyline in space approximating the "ideal" trajectory of the working tool. Before formulating the problem statement, several important circumstances must be noted.

First of all, the command system of the Kawasaki industrial robot controller includes the $MOVE$ command, the format of which is given as in equation (6):

$$Move(< x1, y1 >)(< x2, y2 >) < velocity > \tag{6}$$

Thus, it is practically possible to generate a robot manipulator program, which consists in translating the description of the polyline into a sequence of $MOVE$ commands, during which each segment of the polyline is assigned one $MOVE$ command.

However, this method of generating a robot manipulator program has significant disadvantages of a fundamental nature. For example, a correctly defined robot manipulator program may correspond to a physically impracticable movement process. As an example, let's consider a program consisting of two $MOVE$ commands that specify the trajectory of the manipulator in the form of a polyline ABC, consisting of two segments AB and BC perpendicular to each other. We will assume that the modulus of the velocity of the working tool of the manipulator v on both segments of the polyline is the same (the direction of the vector $\vec{v}$ will change during the movement, but not the modulus of the vector $v = |\vec{v}|$). Such a program will be formally correct and will be accepted for execution by the controller of the robot manipulator. However, it is obvious that the passage of such a trajectory with a constant modulo velocity is physically impossible, since at point B the acceleration of the working tool must be infinite (the velocity vector cannot change by a jump). Practically, when executing such a program, the manipulator's working tool will stop at point B and, accordingly, the segments AB and BC will not be traversed at a constant speed v set by the manipulator program and, in general, the movement along the segments AB and BC will not be uniform. Obviously, problems of this kind will arise when passing any trajectory in the form of a polyline, whatever the angles between adjacent segments of the polyline.

Two factors can be distinguished that affect the quality of execution of the program synthesized in this way:

1. the maximum value of the angle between the segments of the polyline of the trajectory;
2. the magnitude of the modulus of the speed of movement of the working tool of the manipulator.

Movement along the same trajectory in the form of a polyline will be the more different from a uniform one, the greater the speed of movement of the working tool set by the *MOVE* commands. Thus, the considered «direct» method of generating a robot manipulator program is poorly suited for tasks in which precise compliance with the specified time parameters of the trajectory is required.

An alternative to the method of generating a manipulator program, in which the generated program consists of a sequence of *MOVE* commands, is to use the *CIRCLE* command of the *AS* language, along with the *MOVE* commands. The trajectory of the manipulator in this case will be a curve, which is a sequence of geometric primitives - segments of straight lines and arcs of circles. In conventional mathematical terminology, such a curve can be described as follows: A curve parameterized by a natural parameter is given by the dependence of the radius vector $\vec{r}$ corresponding to the point of the curve on the natural parameter s - the length of the curve arc.

The segment $[0, L]$, where L is the length of the curve, is divided by $n-1$ points $s_1, s_2, \dots, s_{(n-1)}$ into n segments $[0, s_1), (s_1, s_2], \dots, (s_{(n-1)}, L]$.

The sequence of points $s_1, s_2, \dots, s_{(n-1)}$ is assumed to be increasing. The function $\vec{r}(s)$ is given by the relations (7).

$$\vec{r}(s) = \begin{cases} \vec{r}_1(s), s \in [0, s_1) \\ \vec{r}_2(s - s_1), s \in (s_1, s_2] \\ \dots \\ \vec{r}_n(s - s_{n-1}), s \in (s_{n-1}, L] \end{cases} \qquad (7)$$

Moreover, each of the functions $\vec{r}_1(s), \vec{r}_2(s), \dots, \vec{r}_n(s)$ describes one of two types of graphic primitives - a straight line segment (described by functions of the form $\vec{r}(s) = \vec{r}_0 + s \cdot \vec{e}$, where $\vec{e}$ is a direction vector of unit length) or a circular arc (described). Let us formulate the requirements for the approximating curve: continuity. As will be shown below, this requirement is met automatically in the methods we propose;

1. Continuity. As will be shown below, this requirement is met automatically in the methods we propose;

2. Smoothness. Smoothness of the first order is guaranteed, i.e. continuity of the function $\vec{r}'(s) = \frac{d}{ds} r(s)$.

Geometrically, the vector $\vec{r}'(s)$ is a unit length vector tangent to the curve. If the trajectory of the working tool of the manipulator is not a smooth curve, then it is physically impossible to pass such a trajectory at a constant modulo speed, that is, the working tool will stop before the break points, and then accelerate after passing the break point. The requirement of smoothness of the trajectory for the case when the trajectory is a sequence of geometric primitives allows a clear geometric interpretation. Adjacent arcs of circles and segments of straight lines must be conjugated with each other. The procedure for conjugating segments with an arc of a circle is used in our proposed method for generating a manipulator program.

3. Compliance with the specified approximation accuracy. The approximation accuracy is specified by a numerical criterion. The approximation procedure is constructed in such a way that the distance from any point of the trajectory curve to the nearest point of the approximating curve does not exceed this threshold value (approximation accuracy criterion). Thus, the technical task of automatically generating a program for the movement of a robot-manipulator along a given trajectory can be divided into two phases: Building an approximation of the trajectory of a curve, which is a sequence of geometric primitives, and translating the result of the first phase into a sequence of *MOVE* and *CIRCLE* commands of the *AS* language. This phase, the

result of which is a text file containing the program of the manipulator, we called the phase of translation of the description of the sequence of geometric primitives into the program of the robot-manipulator.

When developing software that implements algorithms for automatic generation of a robot manipulator program along a given trajectory, the generation program is divided into two software modules. The output result of the first module is a text file containing a textual description of a sequence of geometric primitives in the form of an s - expression. This text file serves as an input data source for the module for translating a description of a sequence of geometric primitives into a robot manipulator program. The translation module itself consists of an s - expression parser designed to read the input file, and the translation block itself. The software implementation of the parser and the translation unit is a standard task, let's consider in more detail the problem of optimal approximation of a spatial curve by a sequence of geometric primitives and methods of curve approximation by a piecewise linear function and their application to solving the approximation problem.

Thus, an original method for approximating plane curves by a sequence of geometric primitives based on the use of the angular characteristic function of the curve has been developed. This method also uses piecewise linear approximation, namely the method of CAL (Chord and Arc Length) - algorithm. The essence of the CAL algorithm is to select several special points on the curve (dominant points) so that the approximation accuracy criterion is met, the convergence of which is strictly proven [25]. The algorithm can be represented by the following sequence of steps:

1) Starting from any point on the closed curve (at the first point on the open curve), calculate the chord length C and the arc length S of the curve, for each subsequent point, and when the criterion value $\frac{1}{2} \cdot \sqrt{S^2 - C^2}$ is greater than the deviation threshold mark the previous point as dominant.

2) Combine the dominant points by testing each dominant point found - can it be eliminated without exceeding the deviation threshold.

3) Calculate the (parametrized) straight line of least squares for the set of points on the curve enclosed between the last two dominant points.

4) Find the point closest to the last dominant point on the constructed straight line.

5) Select the middle of the straight line segment connecting the point found in the previous step with the last dominant point as the last point of approximation.

With a reasonable choice of the deviation threshold value, the algorithm makes it possible to approximate curves of complex shape with polylines with a small number of links.

Despite the fact that *CAL* algorithm was originally developed for plane curves, it was subsequently applied to problems of approximation of curves in three-dimensional space, since both the algorithm itself and the proof of its convergence do not undergo any changes when moving to the case of approximation of a spatial curve. Thus, *CAL* algorithm had been adapted to solve the approximation of the trajectory of a smooth curve, which is a sequence of segments and arcs of a circle. To do this, it is enough to pair adjacent segments of a polyline with arcs of circles. As can be seen from Fig. 2, the radius of the mating circle is bounded from above by the value R_{max}, determined by the formula (8)

$$R_{max} = min\{l_1, l_2\} \cdot ctg\left(\frac{\varphi}{2}\right) \tag{8}$$

where l_1, l_2 are the lengths of adjacent segments of the polyline, and φ is the angle between adjacent segments. Thus, for each vertex of the polyline, the radius of the mating circle can be selected from the interval $[0, R_{max}]$.

When constructing the algorithm for choosing the interface radius, authors proceeded from the considerations that the choice of the radius is a compromise between two extremes. On the one hand, the choice of the maximum possible value of R_{max} may lead to an unacceptably large approximation error, on the other hand, the choice of an excessively small interface radius may lead to unacceptable restrictions on the speed of movement of the working tool. Let us explain the last statement: when moving along the arc of a circle of radius R with a velocity constant modulo v, the magnitude of the centripetal acceleration $\frac{v^2}{R}$ is inversely proportional to the radius and proportional to the square of the velocity.

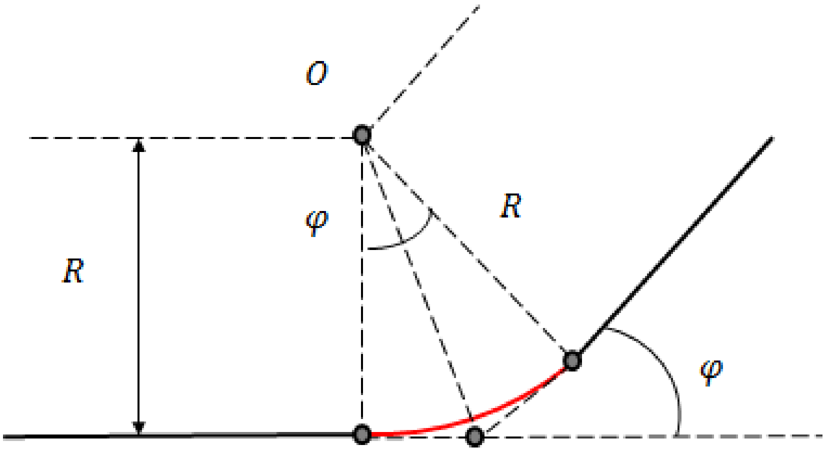

Figure 2. Conjugation of segments by the arc of a circle

It is clear that with a fixed value of the speed of movement, there is a lower maximum permissible threshold of the radius at which the manipulator drive will be able to provide such a trajectory passage. In addition, when moving at a constant speed in a straight line, the acceleration of the working tool is zero, and it is physically impossible to change the amount of acceleration by a jump – when passing the joint, the uniformity of movement is somehow violated. Thus, in order to comply with the time parameters of the trajectory, it is desirable to increase the radius of the mating circle. As a compromise solution, we use an algorithm for choosing the radius of the mating circle, based on calculating the estimate of the approximation error δ by the formula (9)

$$\delta(R) = \frac{\left(1 - \cos\left(\frac{\varphi}{2}\right)\right)}{\cos\left(\frac{\varphi}{2}\right)} \cdot R \qquad (9)$$

At each point of articulation, the value R_{max} is calculated according to the formula (8), then the corresponding error $\delta_{Rmax} = \delta(R_{max})$ is calculated. Then δ_{Rmax} is compared with the specified threshold value of the error δ_{max}. In the case of $\delta_{Rmax} \leq \delta_{max}$, the value R_{max} is chosen as the radius of the mating circle. Otherwise, the radius of the mating circle is calculated by the formula (10)

$$R = \frac{cos\left(\frac{\varphi}{2}\right)}{\left(1 - cos\left(\frac{\varphi}{2}\right)\right)} \cdot \delta_{max} \tag{10}$$

The described algorithm, despite a number of attractive qualities: small computing capacity, guaranteed accuracy of approximation and ease of software implementation, has a significant disadvantage, introducing approximation errors. If we submit a curve representing the arc of a circle to the input of the algorithm, then at the output we will get a sequence of segments conjugated by the arcs of the circle, and not the desired result, that is, a sequence of one element - the arc of the circle coinciding with the input arc.

Thus, an algorithm for optimal approximation of a flat curve by a sequence of geometric primitives: straight line segments and circular arcs has been developed. The algorithm is based on the use of a special mathematical construction introduced by us into consideration – the function of the angular characteristic of the curve.

Let's give a geometric definition of the function of the angular characteristic of the curve. We will determine the position of the point M on the curve by the arc length of the curve AM.

We denote $\vec{e}(s)$ the tangent vector, which is the guiding vector of the line touching γ at the point $M(s)$. Then we can define the function $\alpha(s)$, assuming that for a given arc length s, $\alpha(s)$ is the angle between the vectors $\vec{e}(s)$ and $\vec{v}_0$ (see Fig. 3).

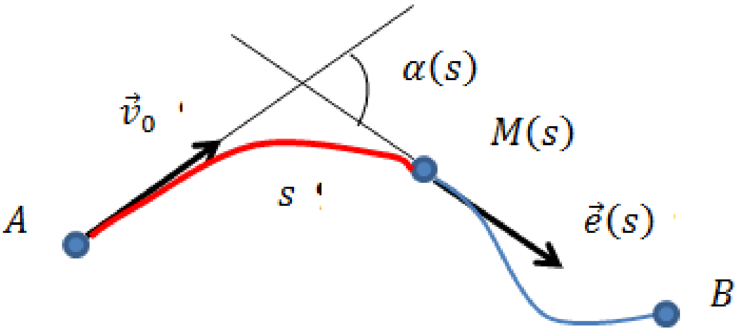

Figure 3. Geometric definition of the function $\alpha(s)$

The function $Ang(\vec{e}_1, \vec{e}_2)$ has been analytically defined on the set of ordered pairs of vectors of unit length $E \times E$, where the set of two-dimensional vectors of unit length $E = \{\vec{e} \in R^2 | \, |\vec{e}| = 1\}$ has been denoted. The range of values of the Ang function will be the segment of the real line $D = [-\pi, \pi]$. For an accurate definition, we introduce the function $S: \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$, defined by the expression (11):

$$S(\vec{a}, \vec{b}) = a_x \cdot b_y - a_y \cdot b_x \tag{11}$$

Geometrically, the function $S(\vec{a}, \vec{b})$ is the oriented area of a parallelogram spanned by the vectors $\vec{a} = (a_x, a_y)$ and $\vec{b} = (b_x, b_y)$. Let $\vec{e}_1, \vec{e}_2 \in E$ and $\vec{e}_1 = (\cos(\alpha), \sin(\alpha))$, $\vec{e}_2 = (\cos(\beta), \sin(\beta))$. Then, obviously, equality (12) is valid:

$$S(\vec{e}_1, \vec{e}_2) = \sin(\beta - \alpha) \tag{12}$$

Now we can define the function Ang by equality (13):

$$Ang(\vec{e}_1, \vec{e}_2) = arcsin(S(\vec{e}_1, \vec{e}_2)) \tag{13}$$

In the future, speaking about the angle α between two nonzero vectors $\vec{a}$ and $\vec{b}$, we will mean the value (14)

$$\alpha = Ang\left(\frac{\vec{a}}{|\vec{a}|}, \frac{\vec{b}}{|\vec{b}|}\right) \tag{14}$$

If the length of the entire curve γ is L (i.e. $s(B) = L$, then the function $\alpha(s)$ is defined on the interval $s \in [0, L]$. Let's define the function $\theta(x)$ by the expression (15) on the interval [0,1]:

$$\theta(x) = \alpha(x \cdot L) \tag{15}$$

We will call $\theta(x)$ the function of the angular characteristic of the curve γ (or simply the angular characteristic γ).

Properties of the angular characteristic function:

6) If the curve γ is a straight-line segment, then its angular characteristic function $\theta(x)$ is determined by the formula (16)

$$\forall x \in [0,1] \;\; \theta(x) = 0 \tag{16}$$

7) If the curve γ is an arc of a circle of radius R, which is cut off by the angle φ, then regardless of the magnitude of R, its angular characteristic function will be a linear function of the form (17)

$$\theta(x) = \varphi \cdot x \tag{17}$$

These properties make it possible to effectively apply the function of the angular characteristic of the curve to the problem of approximating the curve by a sequence of geometric primitives. Namely, if we perform a piecewise linear approximation of the angular characteristic function, then the segments of the polyline will correspond to the arcs of circles and segments, and the adjacent arcs of circles and segments will be conjugate, Thus, the piecewise linear approximation of the angular characteristic function of the curve uniquely corresponds to a smooth curve, which is a sequence of arcs of a circle and segments.

Results and Discussion. Automatic planning of the working tool trajectory and generation of the robotic arm motion program based on 3D scanning data. Thus, the process of constructing the trajectory of the working tool of the robot manipulator developed and tested in this study consists of two stages: at the first stage, the trace of the trajectory is constructed as a list of coordinates of corner points (the dominant points) lying on the corresponding spatial curve, then the actual trajectory of the working tool is constructed, also in the form of a list of coordinates of points. At the same time, for each point of the trajectory trace, a point of the manipulator trajectory is generated by shifting the normal vector to the surface in the direction by an amount that determines the distance from the surface to the tip of the spray cone. Non-working segments (maneuvering segments) are matched with a standard pattern of a sequence of coordinates of points.

The input data for the software module that generates a sequence of commands from the robot manipulator controller is a list of coordinates of sequentially traversed trajectory points. Since the trajectory of the manipulator's working tool is programmatically determined by a sequence of standard movement patterns of two types - movement along a straight line segment (corresponding to the *MOVE* command group of the *AS* language) and movement along a circle segment (corresponding to the *CIRCLE* command group of the *AS* language), the main function of the manipulator program generation module is the optimal translation of a sequence of dominant points of the manipulator trajectory into a text file containing a sequence of commands of the manipulator controller. Note that the easiest-to-implement method of live translation, in which the trajectory of the working tool is a polyline with specified vertices, does not always give satisfactory results. Therefore, for translation, a spline interpolation procedure is used for the input sequence of points, followed by a segmentation procedure for the resulting smooth curve to approximate it with a sequence of geometric primitives.

Thus, authors adapted the *CAL* algorithm to solve the problem of approximating the trajectory of a smooth curve by a sequence of segments and arcs of a circle, and developed an algorithm for optimal approximation of a flat curve by a sequence of these geometric primitives, introducing a special function of the angular characteristic of the curve. The approximation algorithm used in this work for planning the trajectory and generating the program of the robotic arm has guaranteed approximation accuracy and ease of software implementation. It is practically important that the algorithms ensure smooth movement of the working tool of the robot manipulator with a constant modulo velocity along a smooth curve - a 3D model of the product, without the risk of undesirable large values of centripetal acceleration during manipulator maneuvers. As far as the authors know, the application of trajectory planning and program generation algorithms to the architecture of the AS language that controls the Kawasaki robotic arm does not appear in the literature. The algorithms have been used to control the robot manipulator when performing technological operations of plasma cutting and microplasma coating spraying.

To work out technological solutions at the pilot production site, the 3D scanning scheme [12]-[16] and algorithms for segmentation, trajectory planning and generation of the robot program have been used, that is, an intelligent system of robotic plasma cutting and surface treatment has been implemented. During the operation of the system, the robot manipulator sequentially scans the surface with a fixed distance sensor, recreates a 3D model of the surface and performs either plasma cutting (with a fixed plasma cutter) or plasma spraying of the coating (with a fixed microplasmotron), moving along the recreated 3D trajectory, while the robot manipulator itself performs the generation of the motion program (intelligent control).

After testing these algorithms on a pilot production site, new technological solutions were developed for plasma cutting of large-sized products and for spraying protective microplasma coatings in order to restore worn-out large-sized parts, as well as for microplasma spraying of biocompatible coatings.

Dominant Points and Machine Vision. We emphasized in the previous sections the importance of such objects as dominant points in the process of controlling the robot manipulator. Let's give examples of dominant points on the surface:

– pulse glowing dots, light indicators, minilamps;

– round, triangular or square spots (blobs), whose centers can be selected as virtual dominant points;

– vertices of corner structures where corners can have shapes $L, V, T, Y, X,$ i.e. intersection points of some lines, or points of convergence of two dominant boundaries. This can be called a smooth corner, or a curve connecting the ends of two beams;

– segments and their end or midpoints, etc.

As a result, the following issues arise - blob detection, corner detection, and line detection. In this section, we consider the problem of detecting corner points in a hypothetical formulation of the equipment of the robot manipulator with a machine vision system. Equipping a manipulative robot with a technical vision system is one of the possibilities for sensing the robot.

The advantages of a robot equipped with machine vision system are obvious:

– improvement of personnel safety and control of the robot workspace;

– the possibility of complicating the tasks to be solved and increasing the efficiency of the robot;

– using detected dominant (anchor, corner, etc.) points can improve the connectivity of the trajectory and its smoothness;

– additional possibilities of motion programming and part gripping.

Machine vision is the application of computer vision for industry and production. While computer vision is a general set of techniques that allow computers to see, the field of interest in technical vision, as an engineering branch, is digital input /output devices and computer networks designed to control manufacturing equipment, such as robotic arms. Technical vision is a subsection of engineering related to computing, optics, mechanical engineering and industrial automation. Vision systems use digital smart cameras, as well as image processing software to perform similar checks.

Vision systems are programmed to perform highly specialized tasks, such as counting objects on the conveyor, searching for surface defects, controlling movement and actions (welding, cutting, spraying, etc.), and controlling the space safe for personnel near the robot manipulator. Vision systems must "see" by examining the individual pixels of an image, processing them and attempting to draw conclusions using a knowledge base and a set of image processing and pattern recognition functions. Some vision algorithms have been developed to mimic human visual perception. The software includes many algorithms such as binarization, segmentation (search and counting details), checking an image for individual blobs of related pixels as image anchor points. These blobs often represent targets for part processing, capture, or manufacturing rejects.

The technical aspects of pairing the robotic arm and the vision system can be found in more detail in the literature [26]. In this section, we will focus on the detection of dominant points, namely corner points. We review recently developed scalable corner detection masks with flexible geometry and hierarchical structure.

Scalable masks. Depending on the applications, the corner itself is also called the vertex of the angle, that is, a separate point, and a less local object, including, in addition to the vertex, also the rays propagating from it, as well as the entire angular structure. In the processing of three-dimensional 3D images, the sides that make up the border of the corner (generally, a polytop) are added to the straight lines (edges), in which the dominant changes in brightness are visually observed, characterizing the difference between one area of the image (corner) from another (background). One of the common approaches to finding corners consists in boundary detection and binarization, and subsequent detection procedures on a binary analogue of an image. This method is based on the studying the brightness of the image in the vicinity of a point for the equality to zero of the second derivative and a change in the sign in the direction normal to the boundary.

In this paper we consider the group of algorithms that does not perform edge selection and binarization, but works directly with a grayscale image, scanning its elements with a local neighborhood and calculating the correlation of a snapshot fragment with a mask programming the angular structure model (Fig. 4). It is assumed that the inner region of the corner is approximately a plateau. The size of the mask is uneven; when the image is scanned, the central element of the mask is placed in the center of the examined image fragment. For each element of the image, the values of the convolution of the fragment and the mask are calculated for its various rotations about the central element. The maximum absolute value of these is retained as a measure of the presence of a corner at a point.

Many differentiating masks, or discrete kernels of two-dimensional convolution are known

[27]. Among such schemes for constructing masks, the Kirsch mask, which simulates oriented boundaries, stands out:

$$K^1 = \begin{matrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{matrix}, \quad K^2 = \begin{matrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{matrix}, \quad K^3 = \begin{matrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{matrix}, \dots, \quad K^8, \qquad (18)$$

where three of eight rotated versions are shown. The considered detectors have property of non-scalability, which creates problems for organizing fast computations. For example, scanned data with a 3 × 3 mask is problematic to use in calculations with large masks. Calculations with masks of sequentially increasing sizes contain information about the linear and areal parameters and the moment of the corner transition to the background area. In comparison with the Kirsh masks (18), a scalable masks are obtained under the assumption that the border between the corner structure and the background passes inside a certain pixel through its center, and not between two adjacent pixels along their sides or edges. Scalable 3 × 3 masks are shown for comparison with (18):

$$M_2^1 = \begin{matrix} 1 & 3 & 1 \\ -1 & 0 & -1 \\ -1 & -1 & -1 \end{matrix}, \quad M_2^2 = \begin{matrix} -1 & 1 & 3 \\ -1 & 0 & 1 \\ -1 & -1 & -1 \end{matrix}, \quad M_2^3 = \begin{matrix} -1 & -1 & 1 \\ -1 & 0 & 3 \\ -1 & -1 & 1 \end{matrix}, \dots, M_2^8, \qquad (19)$$

where M_n^r denotes the $r-th$ rotated version of the mask ($r = 1, \dots, R$) of $N \times N$ size with $N = 2n - 1$. Let us outline in short the main priciples of scalable masks modeling.

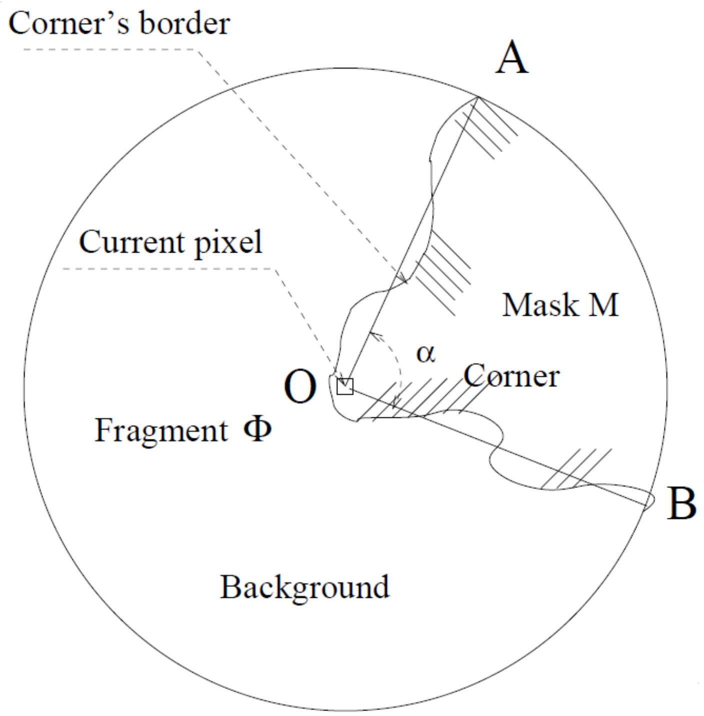

Figure 4. Principal scheme of corner approximation within a circular window. Fragment Φ is located on the image field with a point O in the center. The corner structure approximately has a vertex at the point O. The approach to finding the corner is in the rotation of the ideal angle $< AOB = \alpha$ $(mask\ M)$ and the

calculation of the difference $||\Phi - \mathrm{M}||$

Firstly, the matrix is introduced with a possibility of expanding it to larger sizes. The vertex (0) is centered in the middle of the mask, the corner edges $(e - s)$ embrace the angular body $(c - s)$:

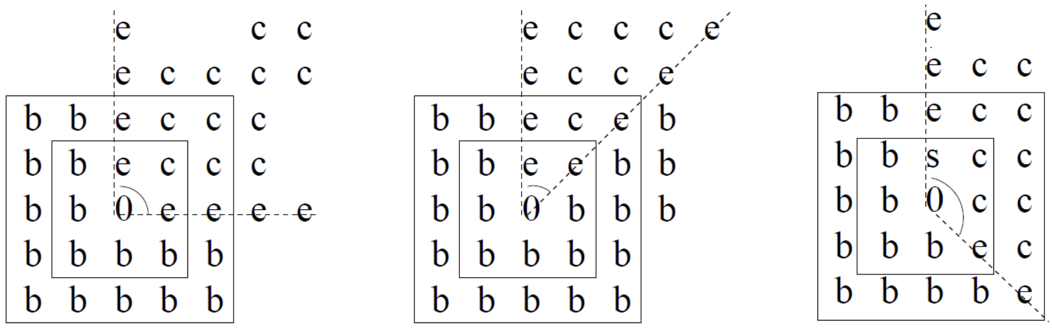

Figure 5. Digital corner models. From left to right: corner of 90, 45, and 135 degrees

$$M_3^1 = \begin{array}{|c|c|c|c|c|}\hline \mathbf{e} & \mathbf{c} & \mathbf{c} & \mathbf{c} & \mathbf{e} \\ \hline b & \mathbf{e} & \mathbf{c} & \mathbf{e} & b \\ \hline & b & \mathbf{0} & b & \\ \hline & & b & & \\ \hline & & & & \\ \hline \end{array}, M_3^2 = \begin{array}{|c|c|c|c|c|}\hline & b & \mathbf{e} & \mathbf{c} & \mathbf{c} \\ \hline & b & \mathbf{e} & \mathbf{c} & \mathbf{c} \\ \hline & b & \mathbf{0} & \mathbf{e} & \mathbf{e} \\ \hline & b & b & b & b \\ \hline & & & & \\ \hline \end{array}, M_3^3 = \begin{array}{|c|c|c|c|c|}\hline & & & & \mathbf{1} \\ \hline & & & \mathbf{1} & \mathbf{3} \\ \hline & & \mathbf{0} & \mathbf{3} & \mathbf{3} \\ \hline & & & \mathbf{1} & \mathbf{3} \\ \hline & & & & \mathbf{1} \\ \hline \end{array}, \ldots, \qquad (20)$$

where empty cells are assigned with background values $(b - s)$. We need some definitions (Fig. 5).

Definition 1 The matrices M_n of the odd size $(2n - 1)^2$, $n = 2, \ldots$ with zero central entry $M_n(n, n) = 0$ and non central entries e, c and b are called scalable masks of angular structures provided the following conditions are satisfied:

– A set of entries $E_n = \{e\}$ simulating the edges of the corner consists of two digital halfines starting at zero point $O = \{0\}$ and spreading from it either along a column and a row, or along a column/row and one of the four diagonals, or along two diagonals.

– A set of entries $C_n = \{c\}$ of the matrix enclosed by the sides of the corner is called a body of the corner.

– The remaining entries form a set $B_n = M_n \setminus (O_n \cup E_n \cup C_n)$, called a background.

The model of the corner mask includes the principle of self-similarity consisting in the fact that, as the number n increases, the values e of the corner sides are extended along a given propagation line (rows, columns, diagonals). Moreover, the values c of the internal elements of the corner body are spreading from the center of the mask to its periphery, whereas the background values b fill the mask region complementary to the corner elements. The central entry of the mask is chosen to be zero, as well as the sum of all mask entries. This explains the differentiating effect on the image, produced by convolution with a sliding mask of this type. Denote by $|E_n|, |C_n|, |B_n|$ and $|O_n| \equiv 1$ the number of mask entries with the values e, c, b and 0, respectively. The number of all mask entries M_n (for convenience, the superscripts are omitted) equals $|M_n| = (2n - 1)^2$. Let us calculate the values $|E_n|, |C_n|, |B_n|$ and write down the differential mask condition as follows. We can express the values$|E_n|, |C_n|, |B_n|$ in the general form

$$|E_n| = 2(n-1), \quad |C_n| = (n-1)^2, \quad |B_n| = (n-1)(3n-1) \tag{21}$$

and write down the differential mask condition as follows

$$|E_n|e + |C_n|c + |B_n|b = 0 \tag{22}$$

Then we arrive at the differential condition (22) in the form:

$$2e + (n-1)c + (3n-1)b{=}0 \tag{23}$$

Due to the scalability of matrices, we have another similar equation for an arbitrary m

$$2e + (m-1)c + (3m-1)b{=}0 \tag{24}$$

and, subtracting (24) from (23), obtain

$$c + 3b{=}0 \tag{25}$$

Then we substitute $c = -3b$ into (23) and find the solution in terms of b:

$$(e, c, b) = (-b, -3b, b) = b(-1, -3, 1) = -b(1, 3, -1) \tag{26}$$

Relatively prime weights $(e, c, b) = (1, 3, -1)$ constitute a scalable mask of the angles of 90 degrees. Scalable masks, their derivation and numerical experiments with test 2D images are given in [28]. Such masks are derived for angles of 90, 45 and 135 degrees. Here is an example of a scalable mask $7 \times 7 (n = 4)$ of 45 degrees.

$$M_4 = \begin{array}{|c|c|c|c|c|c|c|}
\hline
-1 & \vdots & -1 & \mathbf{3} & \mathbf{7} & \mathbf{7} & \mathbf{3} \\
\hline
\vdots & -1 & -1 & \mathbf{3} & \mathbf{7} & \mathbf{3} & -1 \\
\hline
\vdots & & -1 & \mathbf{3} & \mathbf{3} & -1 & -1 \\
\hline
\vdots & -1 & -1 & \mathbf{0} & -1 & -1 & -1 \\
\hline
\vdots & \ldots & -1 & -1 & & & -1 \\
\hline
\vdots & -1 & & \ldots & & -1 & -1 \\
\hline
-1 & & & \vdots & & & -1 \\
\hline
\end{array}$$

Description of corner detection algorithm. Let $\Phi = \Phi_{ijk}$ denote the part of the larger image f, (i.e., a fragment with central pixel (i, j, k)) of $N \times N$ size (in 2D case) and $N \times N \times N$ (in 3D case), on which the sliding window M_n $(N = 2n - 1)$ of the same size as Φ has stopped at a pixel (i, j, k). A measure of similarity of a fragment and a rotational version $M_n^r, r = 1, \ldots, R$ of the mask M_n can be obtained from the well-known identity

$$||\Phi = M_n^r||^2 = ||\Phi||^2 - 2\langle \Phi, M_n^r \rangle + ||M_n^r||^2 \tag{27}$$

in the chosen norm of the Hilbert space. It follows from this relation that the fragment Φ is best approximated by the ideal corner mask M by finding a maximum of the scalar product $\langle \Phi, M_n^r \rangle$, due to $||M_n^r||^2 = const$. Then the criterion for detecting a corner can be formulated in the form

$$Q = \max_{r,n} \langle \Phi, M_n^r \rangle \tag{28}$$

When the fragment Φ of N^3 size, centered in the pixel with the current coordinates (i, j, k) moves over the image field f, the scalar product of the fragment Φ with the matrix of the mask $M_n^r, r = 1, \dots, R$ is computed:

$$u_n^r(i, j, k) = \langle \Phi_{ijk}, M_n^r \rangle \tag{29}$$

We calculate the image of the maximum responses among the rotations

$$U_n^R(i, j, k) = \max_{r=1,\dots,R} |u_n^r(i, j, k)| \tag{30}$$

It is expected that the maximum response occurs with the orientation of the mask that is in the best agreement with the rotation of the corner observed in the fragment. The spatial extension of the corner along its sides can be evaluated by varying another parameter, the mask size n. To this end, we choose the integer Z as an estimate of the maximum size of the corner encountered, and calculate the image of maximum responses to the mask growth:

$$C_Z^R(i, j, k) = \max_{n=1,\dots,Z} U_n^R(i, j, k) \tag{31}$$

It is assumed that, with increasing the tested corner size, the response grows to the limits of the corner extension, and upon reaching them, some saturation of the response or a more complex event occurs due to the mask capturing the non-corner areas. While studying the visual properties of the image C_Z^R, we observe a version of the original image with increased brightness at the corner points.

Now the corner points themselves can be obtained by introducing a classification threshold, which leaves the significant, or dominant points in the image. Many threshold search methods are available, including the dynamic, locally adapted thresholds [27], [28]. The problem of normalization of the detection criterion remains challenging because the corners in images have different intensities. As a consequence, the scatter in the range of the response values C_Z^R in (31) forms the basis for the terms of «strong» and «weak» corners. One of the options for selecting corners is to choose a given number of strong (weak) corners. It appears that the solution to these problems requires a complete description of the corner features in the domain of the parameters (r, n, Z) and time-consuming operations nonlinear with local extremes.

It is known that some information is lost while performing the algorithm steps in (29)-(31). The hierarchical scalable approach we are developing suggests preserving results at these stages as features and examining the data as a whole, not only by the coordinate search for maxima over r and n. The algorithm allows us to sort the corner structures and then to analyze them according to various characteristics, for example, in terms of distributions of orientations of the corners.

The sum of the entries of the mask boundary equals zero. It means that, in addition to the property of masks to have a differentiating character in general (the sum of the entries is zero), the mask boundary also possesses this property. When the size of the growing mask exceeds the corner region and the boundaries of the mask reach the non-corner areas of the image with arbitrary values, the contribution of these regions to the criterion values can change, and a discord is observed. It is difficult to choose in advance the size of a mask in proportion to the size of the desired corner structures, and we encounter the problem of detecting the moment of a significant event (jump, disorder, saturation, etc.) and changing the criterion.

$V - line$ corner masks. Scalable masks M are obtained under the assumption that the body of the angle is a two-dimensional object and fills the triangle area which consists of the proper angle C_n and the boundary E_n. In the images, however, there are also angles in the form of two rays (stripes) with a common vertex, we will denote them V. You can also say that the angle there is a broken $V - line$ [29]. The problem arises - is it possible to build scalable masks V in the same way as we did for M. Define the $V - mask$ model as follows (32)

$$V_3^1 = \begin{array}{|c|c|c|c|c|}\hline \mathrm{s} & b & b & b & \mathrm{s} \\ \hline b & \mathrm{s} & b & \mathrm{s} & b \\ \hline & b & \mathbf{0} & b & \\ \hline & & b & & \\ \hline & & & & \\ \hline \end{array}, V_3^2 = \begin{array}{|c|c|c|c|c|}\hline & b & \mathrm{s} & b & b \\ \hline & b & \mathrm{s} & b & b \\ \hline & b & \mathbf{0} & \mathrm{s} & \mathrm{s} \\ \hline & b & b & b & b \\ \hline & & & & \\ \hline \end{array}, \ldots, \tag{32}$$

Denote by $|S_n|, |B_n|$ and $|O_n| \equiv 1$ the number of mask entries with the values s, b and 0, respectively. The number of all mask entries V_n equals $|V_n| = (2n-1)^2$. Let us calculate the values $|S_n|, |B_n|$ and write down the differential mask condition as follows. We can express the values $|S_n|$and $|B_n|$ in the general form

$$|S_n| = 2(n-1), \quad |B_n| = (2n-1)^2 - 2(n-1) - 1 = 2(n-1)(2n-1) \tag{33}$$

and write down the differential mask condition as follows

$$|S_n|s + |B_n|b = 0 \tag{34}$$

Then we have

$$2(n-1)s + 2(n-1)(2n-1)b=0 \tag{35}$$

And

$$s = -b(2n-1) \tag{36}$$

Assuming background values $b = -1$, receive $s = (2n-1)$, see example (37) for $n = 2$ below:

$$V_2^1 = \begin{array}{|c|c|c|}\hline -1 & \mathbf{3} & -1 \\ \hline -1 & \mathbf{0} & \mathbf{3} \\ \hline -1 & -1 & -1 \\ \hline \end{array}, V_2^2 = \begin{array}{|c|c|c|}\hline -1 & -1 & \mathbf{3} \\ \hline -1 & \mathbf{0} & -1 \\ \hline -1 & -1 & \mathbf{3} \\ \hline \end{array}, V_2^3 = \begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & \mathbf{0} & -1 \\ \hline \mathbf{3} & -1 & \mathbf{3} \\ \hline \end{array}, \ldots, V_2^8, \tag{37}$$

and (38) $n = 3$:

$$V_3 = \begin{array}{|c|c|c|c|c|}\hline & & & -1 & \mathbf{5} \\ \hline & & -1 & \mathbf{5} & -1 \\ \hline -1 & -1 & \mathbf{0} & -1 & -1 \\ \hline & & -1 & \mathbf{5} & -1 \\ \hline & & & -1 & \mathbf{5} \\ \hline \end{array}, \tag{38}$$

Obviously, the resulting V masks are not scalable, since they explicitly depend on the parameter n as it is seen in (36). However, they can be modified and become scalable, so that with increasing matrix sizes contained smaller submatrices unchanged. For this you need increase the perimeter values s a little so that the sum of the perimeter becomes zero, how it works for scalable corner masks M (39).

$$V_3^S = \begin{array}{|c|c|c|c|c|}\hline & & & -1 & \mathbf{7} \\ \hline & & -1 & \mathbf{3} & -1 \\ \hline -1 & -1 & \mathbf{0} & -1 & -1 \\ \hline & & -1 & \mathbf{3} & -1 \\ \hline & & & -1 & \mathbf{7} \\ \hline \end{array}, V_4^S = \begin{array}{|c|c|c|c|c|c|c|}\hline -1 & \dots & -1 & 1 & -1 & -1 & \mathbf{11} \\ \hline \vdots & -1 & -1 & -1 & -1 & \mathbf{7} & -1 \\ \hline \vdots & & -1 & -1 & \mathbf{3} & -1 & -1 \\ \hline \vdots & -1 & -1 & \mathbf{0} & -1 & -1 & -1 \\ \hline \vdots & \dots & -1 & -1 & \mathbf{3} & -1 & -1 \\ \hline \vdots & -1 & & \dots & & \mathbf{7} & -1 \\ \hline -1 & & & \dots & & & \mathbf{11} \\ \hline \end{array} \tag{39}$$

Conclusions

1. A new algorithm to control a robotic arm has been developed, in which automatic trajectory planning and the formation of a motion program for the robotic arm are carried out according to 3D scanning data of the surface of the product being processed by the robot.
2. A new algorithm has been developed for generating a robot program and ensuring smooth movement of the working tool of a robot manipulator with a constant modulo speed along a smooth curve - a 3D model of the product, without the risk of undesirably large values of centripetal acceleration during manipulator maneuvers.
3. The practical implementation of the developed algorithms to control a robot manipulator has been carried out. The algorithms for automatic trajectory planning and generation of the Kawasaki robotic arm program in the AS language are used to control the manipulator that performs the technological operations of plasma cutting and microplasma spraying of coatings.
4. The use of M and V masks in practice requires extensive computational experiments to adapt and possibly combine them. This is the subject of future research.

Acknowledgement. This research was supported by the Committee for Control in the Sphere of Education and Science of the Republic of Kazakhstan for the project AP13068317 «Development of new robot-manipulator control algorithms for 3D scanning technologies and additive microplasma spraying of coatings». The author Kazantsev I.G. was partially supported in the Framework of the State Assignment for the Institute of Computational Mathematics and Mathematical

Geophysics of the Siberian Branch of the Russian Academy of Sciences (project no. 0315-2016-0003).

References

1. Nonlinear control systems – A brief overview of historical and recent advances. Iqbal J, Ullah M., Khan S. G., Khelifa B., Ćuković S. // Nonlinear Engineering. - Vol. 6, Issue 4, pp. 301-312, August 2017. https://doi.org/10.1515/nleng-2016-0077
2. A Tutorial on Robust Control, Adaptive Control and Robust Adaptive Control – Application to Robotic Manipulators. Bin Wei // Inventions. - Vol. 4(3):49. August 2019. http://dx.doi.org/10.3390/inventions4030049
3. Decomposed dynamic control for a flexible robotic arm in consideration of nonlinearity and the effect of gravity. Haibin Yin, Yongguang Li, Junfeng Li. // Advances in Mechanical Engineering. - Vol. 9, Issue 2, pp. 1-15. February 2017. DOI https://doi.org/10.1177/1687814017694104
4. Adaptive Neural Tracking Control of Robotic Manipulators with Guaranteed NN Weight Convergence. Jun Yang, Jing Na, Guanbin Gao , and Chao Zhang. // Complexity. – Vol. 2018, Article ID 7131562, 11 pages, https://doi.org/10.1155/2018/7131562
5. Two-Link Robot Manipulator: Simulation and Control Design. Baccouch M, Dodds S. A. // International Journal of Robotic Engineering. - Vol. 5 (2). December 2020. DOI: 10.35840/2631-5106/4128
6. Wavelet Adaptive Control for Robotic Manipulator with Input Deadzone. Sun Y., Zhao D., Wang Yu. // Journal of Physics: Conference Series. - Vol. 1607(1):012030. August 2020. DOI: 10.1088/1742-6596/1607/1/012030
7. Modeling and Simulation of a Point to Point Spherical Articulated Manipulator Using Optimal Control. Saraf P., Ponnalagu R.N. // Conference: 2021 7th International Conference on Automation, Robotics and Applications (ICARA), February 2021. DOI: 10.1109/ICARA51699.2021.9376496
8. MPC Control and LQ Optimal Control of A Two-Link Robot Arm: A Comparative Study. Guechi El-H., Bouzoualegh S., Zennir Y., Blažic S. // Machines. - Vol. 6(3), pp. 2-14. August 2018. DOI:10.3390/machines6030037
9. Modern Control Laws for an Articulated Robotic Arm: Modeling and Simulation. Iqbal J., // Engineering, Technology & Applied Science Research. - Vol. 9, Issue: 2, pp: 4057-4061, April 2019. https://doi.org/10.48084/etasr.2598
10. Control of an Anthropomorphic Manipulator using LuGre Friction Model - Design and Experimental Validation. Khurram A., Adeel M., Israr M., Sohail R., Jamshed I. // Strojniški vestnik - Journal of Mechanical Engineering. - Vol. 67, Issue 9, pp: 401-410, September 202. https://doi.org/10.5545/sv-jme.2021.7258
11. Nonlinear Control of a Flexible Joint Robotic Manipulator with Experimental Validation. Waqar A., Adeel M., Khurram A., Usman J., Soltan A., Jamshed I. // Strojniški vestnik - Journal of Mechanical Engineering. – Vol. 64, Issue 1, pp. 47-55, January 2018. https://doi.org/10.5545/sv-jme.2017.4786.
12. Development of Control System for Robotic Surface Tracking. Alontseva D. L., Ghassemieh E., Krasavin A. L., Shadrin G. K., Kussaiyn-Murat A. T., Kadyroldina A. T. // International Journal of Mechanical Engineering and Robotics Research. - Vol. 9, No. 2, February 2020. – pp. 280-286. doi: 10.18178/ijmerr.9.2.280-286
13. Development of 3D Scanning System for Robotic Plasma Processing of Medical Products with Complex Geometries. Alontseva D. L., Ghassemieh E., Krasavin A. L., Kadyroldina A. T. // Journal of Electronic Science and Technology. – Vol. 18(3), 2020. – Pp. 212-222. doi: 10.1016/j.jnlest.2020.100057
14. Segmentation Algorithm for Surface Reconstruction According to Data Provided by Laser-Based Scan Point. Alontseva D., Krasavin A., Kadyroldina A., Kussaiyn-Murat A. // Communications in Computer and Information Science. - Vol. 998, 2019. - pp, 1-10. https://doi.org/10.1007/978-3-030-12203-4_1.
15. Development of an industrial robot manipulator control system for three-dimensional scanning of surfaces. Alontseva D.L., Krasavin A.L., Shadrin G.K., Kadyroldina A.T., Kusayyn-Murat A.T. // D.Serikbayev EKSTU Bulletin. No. 1, 2019. - C. 81-87(in Russian).
16. Development of an information system for a robot manipulator performing plasma processing of complex products. Kadyroldina A.T., Kusain-Murat A.T., Krasavin A.L., Prokhorenkova N.V. // D.Serikbayev EKSTU Bulletin, No. 3 (89), 2020, pp 95-98 (in Kazakh).

17. Investigation of the deposition spot and the pattern of metallization under the conditions of microplasma deposition of a coating of titanium dioxide]. Borisov Yu.S., Voinarovych S.G., Kyslytsia O., Kalyuzhnyj S. N. // Automatic welding. – 2014. – № 12. - pp. 19-21 (in Russian).
18. Automatic Spray Trajectory Optimization on Bézier Surface. Chen W., Liu J., Tang Y.,Ge H. // Electronics. 2019. – Vol. 8 (168). – P. 1-16.
19. Uniform coverage of automotive surface Patches. Prasad N.A., Aaron G., Conner C.D., Choset H., Rizzi A. A. // The International Journal of Robotics Research. - 2005. - Vol. 24, № 11. – P. 883-898.
20. Trajectory planning and control for robot manipulations: doctoral thesis. Zhao R. 24.09.2015 / Université Paul Sabatier. – Toulouse, 2015. - 158 p.
21. Motion Control of Industrial Robots in Operational Space: Analysis and Experiments with the PA10 Arm. Campa R., Ramirez C., Camarillo K., Santibanez V., Soto I. // Advances in Robot Manipulators, Ernest Hall (Ed.). – InTech, 2010. – P. 417-442.
22. A New Adaptive Tracking Control Approach for Uncertain Flexible Joint Robot System. Liu Zh.G, Huang J.M. // International Journal of Automation and Computing. - 2015. – Vol. 12(5). – P. 559-566.
23. Research on Robot Surface Tracking Motion Based on Force Control of Six-Axis Wrist Force Sensors. Wang Zh., Liu W., Cui B., He J., Li Zh., and Zhao Yo. // Advances in Mechanical Engineering. – Vol. 3. – 2014. – P.1-9. – Article ID 249696
24. Lectures on Differential Geometry. Chern S.S., Chen W.H., Lam K.S. // World Scientific. – 1999. – 368 p.
25. Efficient piecewise linear approximation of space curves using chord and arc length. Horst J.A., Beichl I. // Proceedings of the SME Applied Machine Vision ‘96 Conference. – Cincinnati Ohio, 1996. – P. 1-12.
26. Robot Guidance Using Machine Vision Techniques in Industrial Environments: A Comparative Review. Perez L., Rodriguez I., Rodriguez N., Usamentiaga R., Garcia, D.F. Sensors 2016, 16, 335.
27. Digital Image Processing. Gonzalez R.C., Woods R.E. // Pearson. – 2018. – 1022 p.
28. Detection of the Corner Structures in Images by Scalable Masks. Kazantsev I.G., Mukhmedzhanova B.O., Iskakov K.T., Mirgalikyzy T. // Journal of Applied and Industrial Mathematics. - 2020. - Vol. 14. – P. 73-84.
29. Inversion of the V-line Radon transform in a disc and its applications in imaging. Ambartsoumian G. // Computers & Mathematics with Applications. – 2012. – Vol. 64. – P. 260-265.