SOLUTION TO THE PROBLEM OF DESIGNING A SAFE CONFIGURATION OF A HUMAN UPPER LIMB ROBOTIC PROSTHESIS

Bureev ASh[1,2], Golobokova EV[1,2], Zhdanov DS[1,2] ✉, Kosteley YaV[1,2], Koshelev RV[2], Seleznev AI[1,2], Fomenko EA[2]

[1] Center for Development of Science, Technology and Education in the Field of Defense and State Security, National Research Tomsk State University, Tomsk, Russia

[2] Federal Research and Clinical Center of Medical Rehabilitation and Balneology of the Federal Medical Biological Agency, Moscow, Russia

This study aimed to develop a method allowing to improve safety of use of robotic medical rehabilitation devices by designing and testing an algorithm for calculation of the angular positions of rehabilitation robotic manipulators or robotic prostheses and allowing to reproduce the natural arc of a human arm under control of a CVS. The Introduction section supports the urgency of development of the methods granting control over positioning of robotic manipulators with the help of a computer vision system (CVS) and thus guarantee safety of patients and medical personnel in the context of work with medical robotic rehabilitation devices. The Materials and Methods section contains a brief description of the robotic arm used in this study, a description of the existing approaches to calculation of angular positions of drives, and a description of the proposed algorithm. The final sections compare application of the proposed algorithm and existing methods of calculation of angular positions of drives of robotic manipulators (robotic prostheses) and outline the possible directions for further improvement.

РЕШЕНИЕ ЗАДАЧИ ФОРМИРОВАНИЯ БЕЗОПАСНОЙ КОНФИГУРАЦИИ РОБОТИЧЕСКОГО ПРОТЕЗА ВЕРХНЕЙ КОНЕЧНОСТИ ЧЕЛОВЕКА

А. Ш. Буреев[1,2], Е. В. Голобокова[1,2], Д. С. Жданов[1,2] ✉, Я. В. Костелей[1,2], Р. В. Кошелев[2], Е. А. Фоменко[2]

[1] Центр развития науки, технологий и образования в области обороны и обеспечения безопасности государства, Национальный исследовательский Томский государственный университет, Томск, Россия

[2] Федеральный научно-клинический центр медицинской реабилитации и курортологии Федерального медико-биологического агентства, Москва, Россия

На сегодняшний день остается актуальной разработка методов контроля позиционирования роботических манипуляторов с помощью систем технического зрения (СТЗ) с целью обеспечения безопасности пациентов и медицинского персонала при работе с медицинскими роботизированными реабилитационными устройствами. Целью исследования было разработать метод повышения безопасности применения роботизированных медицинских реабилитационных устройств путем разработки и апробации алгоритма расчета угловых положений роботизированных манипуляторов или роботических протезов, применяемых в восстановительном лечении и позволяющих воспроизвести естественную траекторию перемещения руки человека под контролем СТЗ. Дано описание роботизированного манипулятора, использованного при проведении исследований, представлены существующие подходы к расчету угловых положений приводов, а также описание предлагаемого алгоритма. Приведены сравнительные результаты работы предлагаемого алгоритма и существующих методов расчета угловых положений приводов роботизированных манипуляторов (роботических протезов) и предполагаемые направления для его доработки.

The current worldwide trend around the latest R&D achievements involves active introduction of robotic equipment in all sectors of the economy. A paper by The Boston Consulting Group (BCG) states that by 2030 the total global market of professional robotics will reach $260 billion [1].

Medical robotics is one of the leading segments of professional robotics by the level of technology employed and demand present. As Joseph Engelberger, father of commercial robotics, said, "... hospitals are the ideal place and the ideal environment for use of robots" [2].

Nevertheless, despite the broad introduction of robotic medical systems, the matters of safety of patients and medical staff that use such systems have not been investigated sufficiently. Some of the published studies point to the documents reporting results of operations that employed robots, and the number of subsequent adverse consequences exceeded one and a half thousand [3]. During the period from 2000 through 2013, surgeries done with the help of robots resulted in death of 144 people. Between 2000 and 2013, the equipment ignited or failed on more than 190 cases. Almost 800 other cases of adverse consequences of robotics-enabled operations resulted from systemic errors such as loss of the video feed [3].

According to the authors of the article, robotic rehabilitation implies slightly different risks for patients: robot arms may move incorrectly (along an unacceptable trajectory or at an unacceptable angle) and thus injure the patient or medical personnel. With this in mind, we undertook to make use of medical rehabilitation robots safer for people by developing a method installing an additional control loop for the robot manipulator's movements that relies on a computer vision system (CVS). "Robot arm movement control" in the context of this study means establishing the fact of that arm reaching a preset point within a local coordinate system. Thus, we employed CVS only to confirm the successfully performed movements without assessing the controlled manipulator's final configuration. For the stated purpose, we analyzed the 3D coordinates for each manipulator's structural components.

The subject we tested the developed method of positioning robotic manipulator's components on was an AR-600E anthropomorphic robot (NPO Androidnaya Tekhnika; Russia). The testing sought to confirm the possible way to improve safety of use and accuracy of positioning of anthropomorphic robotic prostheses of human upper limbs. We paid special attention to finding a solution to the problem of establishing the coordinates of individual components of the manipulator to design its configuration to mimic movements of a human arm in the best possible way. The solution allows designing missing or dysfunctional arm replacements with kinematics matching kinematics of a human arm as close as possible, which makes them safer in use.

It should be taken into account that, unlike industrial robot arms, an anthropomorphic robotic rehabilitation manipulator is not fixed on a rigid base. Coupled with mechanical complexity and a large number of interconnected components, this translates into considerably inaccurate positioning values, which, in some cases, can make the robotic prosthesis dangerous to its owner or people around him/her. Under such conditions, the task of accurate and safe positioning is not a trivial one; it largely depends on the method of designing the mechanism itself.

A suggested solution to this task involves a CVS module integrated into the robotic rehabilitation anthropomorphic manipulator's control loop. This module would monitor the position of the manipulator in its field of view and generate commands to interrupt or correct a potentially dangerous movement. The module is supposed to recognize and track both the manipulator's grip and special markers attached to its components [4, 5]. However, even with a CVS enhancing the traditional methods of estimation of the manipulator's position (relative to the elements of the CVS), the absolute mean calibration error between the system [6, 7] and the manipulator's grip is more than ± 5 cm [4].

The traditional approach to determining the current position of the manipulator's components involves requesting their 3D coordinates from the direct kinematics logic. The inputs are the values transmitted by the angular position sensors of the corresponding drives. The logic also contains the current coordinates by the CVS. At the initial stages, the software must search for the target object and calculate its spatial position relative to the elements of the system or the absolute zero of the kinematics logic. This can be done with the help of position-based (PBVS), image-based (IBVS) visual servoing or hybrid methods. In general, the above methods calculate the needed coordinates by analyzing images; they are applicable to both industrial manipulators and anthropomorphic robots and robotic prostheses of the human hand.

Image-based visual servoing allows comparing the calculated needed and current positions of the manipulator or object on a plane. The difference between the needed and the current positions (the error) is used as feedback. The connection between the received information and changes in the position of the components is made through a Jacobi matrix and the direct kinematic logic of the robot [8]. There are a large number of methods to determine this connection [9–11]. It should be taken into account that a single marker on the object (either the manipulated object or the manipulator itself) enables control over only two degrees of freedom. At least four markers are needed to control six degrees of freedom. Greater number of markers also increases the probability of an unambiguous decision supporting the control command [11, 12]. The IBVS method does not allow linear control of the robot components and does not rely on 3D information about position of the manipulated object. This leads to generation of non-optimal or unrealizable trajectories, a problem that can be solved through adjustments by selected visual parameters [13–15].

The PBVS method implies the coordinates of objects inside the manipulator operating space are determined relative to the coordinate system of the camera part of the CVS. The parameters of the geometric model of the tracked object and camera parameters are factored in together for this purpose. The parameters of the tracked manipulator in the operating space are known; the changes of these parameters can be tracked by responses from the robot's kinematics logic. Geometrical parameters of the manipulated object, on the contrary, strongly depend on the CVS parameters and the adopted methods of 3D localization [16–22].

Hybrid visual servo-enabled control involves both the IBVS and PBVS methods. This approach improves the accuracy of the generated commands through separation of control over manipulator's degrees of freedom [23–27]. Systems that rely on such an approach are less dependent on the robot cameras' calibration accuracy and give a more true idea of the objects' geometry. However, such systems harder to build and consume more computing resources. Moreover, they do not eliminate the risk of non-optimal or unrealizable trajectories, which can still impair the process due to positioning errors and/ or incorrect estimation of the 3D coordinates of the tracked objects by the CVS.
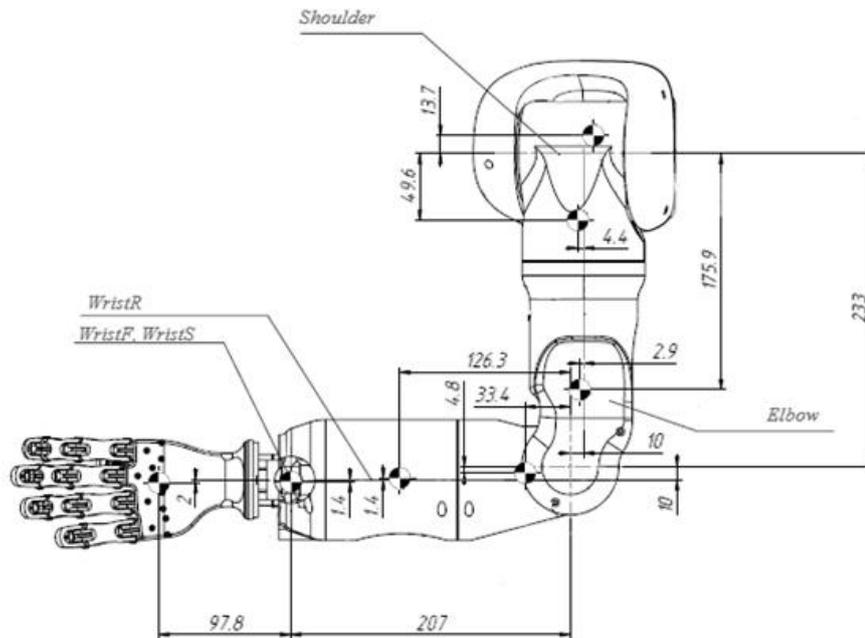
Fig. 1. Structure of the AR-600E manipulator

Calculation of the optimal positions of the manipulator's grip should factor in mechanical limitations of the joints peculiar to both the initial positioning and the subsequent manipulation stages. In addition, since of anthropomorphic prostheses are supposed to be a safer robotic rehabilitation device fir for use in "human" environment, positioning of the manipulator may include stopping before obstacles while moving to target position.

With the current methods of processing data inputs from the CVS, development of the software generating movement trajectories for the AR-600E manipulators does not exclude the possibility of generation of a trajectory (arm to target object) that is either unsafe or unrealizable. In addition, we undertook to enable generation of the trajectories that mimic natural movements of the human arm. In the context of this study, we assessed various options of solutions to the manipulator's inverse kinematics problem and formulated a method that ensures building the movement trajectory as expected.

This study aimed to develop a method making use of robotic medical rehabilitation devices safer by designing and testing an algorithm for calculation of the angular positions of robotic manipulators or robotic prostheses used for rehabilitation purposes and capable of reproducing the natural human.

METHODS

For this study, we developed a number of algorithmic solutions allowing to build a movement trajectory for an anthropomorphic manipulator that closely resembles that of a human arm. The solutions were implemented as software that controls the manipulator of an AR-600E anthropomorphic robot in a simulation environment enabled with quaternion algebra. The manipulator includes the following components (Fig. 1):

1) the Shoulder groups of components moves the Elbow component and the lower components of the manipulator along the frontal (ShoulderS), sagittal (ShoulderF) and vertical axes (ElbowR);

2) the Elbow component moves its child components along the sagittal axis;

3) the WristR component moves its child component along the vertical axis;

4) the WristF and WristS components move the hand along the frontal and sagittal axes, respectively.

The fingers are driven with actuators located on the WristS component.

The Forward and Backward Reaching Inverse Kinematics (FABRIK) method [28] was used to calculate the spatial position of the manipulator's components. This method accounts for the constraints and represents the components of the components, which, combined, allow bringing the hand to the target position in the 3D space.

Two cameras on the head of the anthropomorphic mechanism captured color images within the operating space. Machine learning methods enabled control of the movements of manipulator's grip and tracking thereof, assessment point being whether it has reached the set coordinate in the operating space or not. The main task set before the routines that incorporate machine learning is to detect the grip and assess how it performs a given task within the operating space.

Two approaches to assess correctness of the grip positioning were applied: 1) by requesting responses from the manipulator drives and modeling the current configuration of the manipulator based thereon, and 2) with the CVS of anthropomorphic mechanism. Configuration of the manipulator in its entirety is not controlled, since the CVS' field of view does not cover all of its components. This task is solved after modeling the spatial position of the manipulator, through calculation of angular positions of the drives that bring it to the set 3D coordinates. Below, we consider the possible ways of their calculation on the example of the Shoulder group (Fig. 1) the movements of which affect the spatial position of all other components of the manipulator. The following methods of angular positions calculation were compared:

1) using orthogonal projections of specific points on the manipulator's components. The angle between two points (for example, axis of the Elbow component and the Shoulder group) was calculated by the following formula:

$$angle = 90° - asin\left(\frac{a}{c}\right), \qquad (1)$$

where a is the module of the y coordinate for the corresponding axis of the component, and c is the distance between components in the corresponding orthogonal plane;

2) as angle of rotation of the orthogonal plane between projection of the Elbow component during initialization of the kinematic logic and projection in the analyzed plane of its movement. We used the following formula in this case:

$$cos(\bar{a};\bar{b}) = \frac{(\bar{a};\bar{b})}{|\bar{a}|\bullet|\bar{b}|} = \frac{a_1\bullet b_1 + a_2\bullet b_2 + a_3\bullet b_3}{\sqrt{a_1{}^2 + a_2{}^2 + a_3{}^2}\bullet\sqrt{b_1{}^2 + b_2{}^2 + b_3{}^2}}, \quad (2)$$

where $\bar{a}$ and $\bar{b}$ are the 3D vectors between which the angle is calculated, and a_n, b_n are the corresponding coordinates of 3D vectors after modeling of the orthogonal projections;

3) through the Elbow component's 3D coordinates projection onto the corresponding Shoulder group movement planes:

$$YZ = \left[0; \frac{Elbow.y*dist}{Elbow.x+dist}; \frac{Elbow.z*dist}{Elbow.x+dist}\right],$$

$$XZ = \left[\frac{Elbow.x*dist}{Elbow.y+dist}; 0; \frac{Elbow.z*dist}{Elbow.y+dist}\right],$$

$$XY = \left[\frac{Elbow.x*dist}{Elbow.z+dist}; \frac{Elbow.y*dist}{Elbow.z+dist}; 0\right], \quad (3)$$

$$dist = \sqrt{(Elbow.x - null.x)^2 + (Elbow.y - null.y)^2 + (Elbow.z - null.z)^2}$$

where *dist* is the distance between the current coordinate on the Elbow component axis and the common axis (null) of ShoulderF, ShoulderS components (Shoulder group). We used both (1) and (2) to calculate the angular positions;

4) with the help of the "Shortest arc" method used by game developers to calculate the shortest arc of movement of connected components of virtual objects from the initial point to a given point in 3D space. Practically, implementation by this method means generation of the rotation quaternion from the double rotation and zero rotation quaternions, with the resulting quaternion being the sum of the double rotation quaternion and the identity quaternion;

5) using the algorithm suggested by the authors of this article.

The angular positions calculation method suggested and tested by the authors makes use of a matrix containing interrelated positions of the manipulator components, i.e., a set of coordinates corresponding to certain angles. The increment between them is set during initialization of the algorithm, which prevents repeated generation of the matrix. The input data for the suggested algorithm is the 3D coordinate of a point (center of the hand; Fig. 1) on the manipulator's grip. Next, application of the FABRIK approach yields a set of coordinates of the axis points of the main components' nodes, which, combined, form configuration of the manipulator in 3D space. After that, inside the generated matrix of interrelated positions, using the principle of minimum distances (dist; formula (3)), the most suitable angular position of the drive is selected between the coordinates of the estimated position of the Elbow component and the coordinates stored in the matrix. If necessary, the angular position is adjusted through generation of a local matrix of interrelated positions with a more accurate increment within a small range of rotation angles. Then the angular position of the Elbow component drive is calculated by formula (1), with a being the y axis value by the orthogonal projection of the target position and c being the length of the component. The ElbowR drive's angular position is calculated relative to the axis of the WristR component (Fig. 1). This step involves a number of iterations: sequentially formed quaternions enable rotation of WristR to a given increment. The procedure stops upon reaching the minimum distance between the coordinates calculated by the FABRIK method and the coordinates extracted from the matrix of interrelated positions.
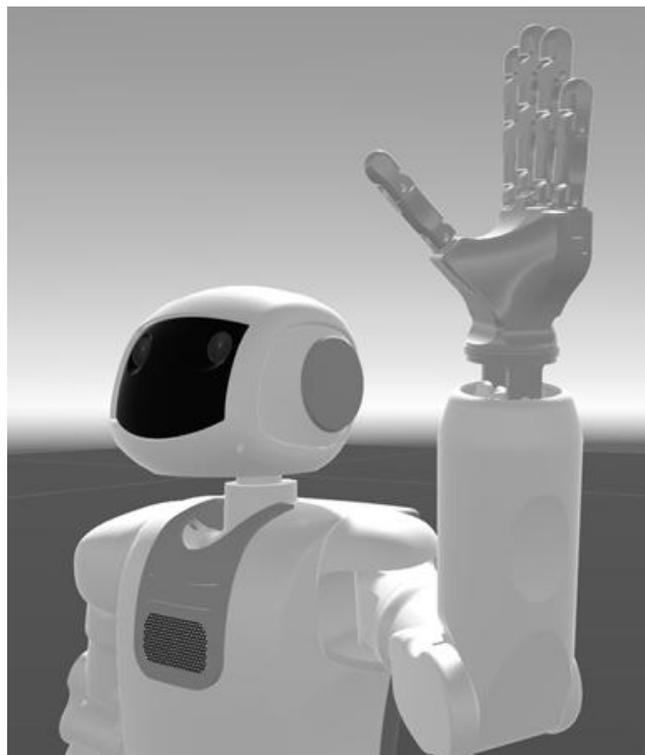

Fig. 2. Example of configuration of the manipulator of AR-600E anthropomorphic mechanism

RESULTS

The experimental part of the study involved application of the considered methods to calculate angular positions of drives of the AR-600E anthropomorphic robot's manipulator in a simulation environment. For this purpose, operator manually set target positions of the components and registered the resulting angular positions of their drives and 3D coordinates of the axes of Elbow and WristS drives. The next step was to compare the results and select the best method by proximity of the resulting coordinates with the coordinates of axes of the components registered by the operator. Figure 2 shows one of the manipulator configurations recorded during the experimental part of the study.

Table 1 shows angular positions of the components of the Shoulder group and Elbow component:

1) 0 — initial position for the anthropomorphic mechanism;

2) 1 — angular positions set by the operator;

3) 2 — angular positions for the configuration calculated by formula (1);

4) 3 – angular positions for the configuration calculated by formula (2);

5) 4 — angular positions for the configuration calculated through the projection of the coordinate onto the respective axis by formula (1);

6) 5 — angular positions for the configuration calculated through the projection of the coordinate onto the respective axis by formula (2);

7) 6 — angular positions calculated by the suggested method.

Elbow and WristS cells of Table 1 contain the 3D coordinates of axes of these components that were discovered through application of the formed angular positions of the drives to them. Cells Elbow and WristR present information about angular positions of the drives (in degrees) that move the respective components to target points in 3D space.

Table 1. Comparison of the results of application of various algorithms enabling calculation of the angular positions of component drives of the anthropomorphic manipulator

	Angular positions of components				Coordinates of components					
Position I										
Method	Elbow		WristS		Elbow			WristS		
	ShF, °	ShS, °	ElR, °	El, °	x, mm	y, mm	z, mm	x, mm	y, mm	z, mm
0	0	0	0	0	191,92	229	3,58	201,85	−14,5	−49,5
1	−26,72	39,16	−3,16	−52,48	321,34	304,77	109,56	394,25	239,27	338,9
2	−27,42	43,16	58,51	−53,52	340,21	318,79	105,39	526	320,28	271,84
3	−27,42	47,72	52,23	−53,52	353,67	330,78	101,59	544,74	347,27	261,05
4	−27,42	43,16	58,51	−53,52	340,25	318,85	105,42	525,96	320,33	271,87
5	−27,42	60,97	52,23	−53,52	586,38	432,61	223,53	387,28	370,68	86,67
6	−26,73	39,15	−2,90	−52,53	321,31	304,78	109,6	394,77	239,73	338,92
Position II										
Method	Elbow		WristS		Elbow			WristS		
	ShF, °	ShS, °	ElR, °	El, °	x, mm	y, mm	z, mm	x, mm	y, mm	z, mm
0	0	0	0	0	191,92	229	3,58	201,85	−14,50	−49,50
1	10,45	2,54	11,73	−84,81	211,85	231,44	−35,88	222,95	116,65	185,26
2	10,52	3,07	20,71	−85,86	215,97	231,91	−36,70	264,48	124,16	182,92
3	10,52	8,61	20,71	−85,86	238,42	234,61	−31,99	292,75	135,74	190,43
4	10,52	3,07	20,71	−85,86	215,97	231,91	−36,70	264,48	124,16	182,92
5	10,52	8,61	20,71	−85,86	238,42	234,61	−31,99	292,75	135,74	190,43
6	10,51	3,04	11,15	−84,89	213,77	231,59	−35,70	223,09	117,25	185,78
Position III										
Method	Elbow		WristS		Elbow			WristS		
	ShF, °	ShS, °	ElR, °	El, °	x, mm	y, mm	z, mm	x, mm	y, mm	z, mm
0	0	0	0	0	191,92	229	3,58	201,85	−14,50	−49,50
1	−70,09	69,21	9,7	−31,31	393,83	443,58	112,32	598,23	493,83	246,09
2	−74,57	85,2	69,97	−32,36	419,98	465,41	50,68	658,75	533,46	72,92
3	−76,68	85,49	68,62	−32,36	421,26	467,23	45,12	660,52	535,69	59,34
4	−74,57	85,2	69,97	−32,36	419,98	465,41	50,68	58,75	533,46	72,92
5	−76,68	86,64	68,62	−32,36	421,5	467,5	44,02	660,75	536,27	57,02
6	−70,00	69,64	8,2	−31,41	394,33	444,03	111,57	598,59	494,73	245,3

Table 2 below illustrates the absolute difference between the coordinates of the Elbow and WristR axes, i.e., the difference between the coordinates resulting from operator's actions (manual movement of the components) and coordinates discovered through application of each of the considered method to calculate angular positions to which the drives moved. Table 2 does not include information on positions 0 and 1.

DISCUSSION

Analysis of the results given in Tables 1 and 2 allows deducing that the suggested drive angular position calculation method ensures generation of the target configuration of the manipulator and movement of its components to the target points. There are also visible differences between the manipulator components' position coordinates calculated by the suggested algorithm and learned as a result of manual movements by the operator. Axis coordinate of the WristS component presents the greatest discrepancy between the obtained results and the reference values. To compensate for this error, we applied an algorithm that, like the one discussed above, uses a matrix of interrelated positions. The WristS component should be moved to its initial position for this purpose. Then, the matrix of interrelated positions is scanned for the drive's angular position value that fits the minimum distances condition. To speed up performance of the algorithm, we decided to calculate the initial value of the WristF drive's angular position:

$$angle = -\left(nullWristF - \left(\pi - acos\left(\frac{a^2+b^2-c^2}{2*a*b}\right)\right)\right), \qquad (4)$$

where *angle* is a precalculated rotation angle of the WristS drive along the sagittal axis; *nullWristF* is the static angle of rotation of WristS relative to WristF, calculated during the initialization of the anthropomorphic robot's software; *a* is the length of the forearm of the anthropomorphic robot; *b* is the distance from the WristF axis to the WristS axis; *c* is the distance from the axis of WristS to the axis of WristR.

We calculated the angular position of the WristR drive in a similar way to the respective calculation for ElbowR. Table 3 presents adjustments of the manipulator configurations shown in Table 1.

The "Difference between deviations" line in Table 3 reflects the magnitude and direction of changes of distances (in millimeters) between the target 3D coordinates of the manipulator components and their values calculated before

Table 2. Absolute difference between the coordinates of the axes of Elbow and WristR components discovered through application of the considered methods to calculate their drives' angular positions

	Angular positions of components				Coordinates of components					
Position I										
Method	Elbow		WristS		Elbow			WristS		
	ShF, °	ShS, °	EIR, °	El, °	x, mm	y, mm	z, mm	x, mm	y, mm	z, mm
2	0,7	4	61,67	1,04	18,87	14,01	4,17	131,75	81,01	67,06
3	0,7	8,56	55,39	1,04	32,33	26	7,97	150,49	108	77,85
4	0,7	4	61,67	1,04	18,91	14,08	4,14	131,71	81,06	67,03
5	0,7	21,81	55,39	1,04	265,04	127,84	113,97	6,98	131,41	252,23
6	0,01	0,01	0,26	0,05	0,03	0,01	0,04	0,52	0,46	0,02
Position II										
Method	Elbow		WristS		Elbow			WristS		
	ShF, °	ShS, °	EIR, °	El, °	x, mm	y, mm	z, mm	x, mm	y, mm	z, mm
2	0,07	0,53	8,98	1,05	4,12	0,47	0,82	41,53	7,51	2,34
3	0,07	6,07	8,98	1,05	26,57	3,17	3,89	69,8	19,09	5,17
4	0,07	0,53	8,98	1,05	4,12	0,47	0,82	41,53	7,51	2,34
5	0,07	6,07	8,98	1,05	26,57	3,17	3,89	69,8	19,09	5,17
6	0,06	0,5	0,58	0,08	1,92	0,15	0,18	0,14	0,6	0,52
Position III										
Method	Elbow		WristS		Elbow			WristS		
	ShF, °	ShS, °	EIR, °	El, °	x, mm	y, mm	z, mm	x, mm	y, mm	z, mm
2	4,48	15,99	60,27	1,05	26,15	21,83	61,64	60,52	39,63	173,17
3	6,59	16,28	58,92	1,05	27,43	23,65	67,2	62,29	41,86	186,75
4	4,48	15,99	60,27	1,05	26,15	21,83	61,64	539,48	39,63	173,17
5	6,59	17,43	58,92	1,05	27,67	23,92	68,3	62,52	42,44	189,07
6	0,09	0,43	1,5	0,1	0,5	0,45	0,75	0,36	0,9	0,79

and after adjustment for the angular positions of the WristS drive. The analysis of the presented data allows deducing that an additional stage of adjustment of the coordinates makes manipulator movements more accurate. In such a case, the movement error does not exceed 0.5 mm. This mechanism can be used when setting a large search increment in generation of the interrelated positions matrices. A more accurate adjustment of the configuration is undertaken at an additional adjustment stage, which ultimately speeds up the suggested algorithm.

CONCLUSIONS

We developed an algorithm for calculation of angular positions of the manipulator components' drives that produces the most accurate and predictable result (Tables 2, 3), which ultimately allows forming the configuration designed by the operator. The algorithm also minimizes the probability of an unpredictable result and robot arm movements along a trajectory unsafe for human beings. It should be noted separately that the accuracy of calculation of the coordinates with the help of the suggested algorithm depends directly on the search increment value (in degrees) set for the interrelated positions matrices generation stage. The algorithm was coded and optimized in C++. Its execution time on a personal computer (Intel Core i7–4770 3.40 GHz, RAM 16 Gb) ranged from 5 to 8 ms, which is sufficient for the purpose of controlling the manipulator of an anthropomorphic robot and a robotic prosthesis. The accuracy of the drives' angular position calculations can be improved by reducing the increment used at interrelated positions matrices generation stage. This, however, would require more RAM capacity for the control software and slow down execution of the algorithm. The way to mitigate this problem is to add a modified version of the suggested algorithm to the control software, i.e. a version that would

Table 3. Results of adjustment of the angular position of WristS drive

Position	Position I			Position II			Position III		
Angular positions, °	WristF	WristS	WristR	WristF	WristS	WristR	WristF	WristS	WristR
	0	0,25	−0,04	0,25	−0,25	0,21	0,25	−1,5	−0,01
Coordinate axis, mm	x	y	z	x	y	z	x	y	z
Target coordinate	394,25	239,27	338,9	222,95	116,65	185,25	598,23	493,83	246,09
Before adjustment	394,77	239,73	338,92	223,09	117,25	185,78	598,59	494,73	245,3
After adjustment	394,93	239,8	338,88	222,08	117,03	185,45	598	494,15	246,13
Deviation along the coordinate axis, mm	x	y	z	x	y	z	x	y	z
Before adjustment	0,52	0,46	0,02	0,14	0,6	0,53	0,36	0,9	0,79
After adjustment	0,68	0,53	0,02	0,15	0,38	0,2	0,23	0,32	0,04
Difference between deviations	−0,16	−0,07	0	−0,01	0,22	0,33	0,13	0,58	0,75

calculate angular positions of the manipulator drives factoring in its previous configuration. In this case, the search in the matrices of interrelated positions will be significantly more narrow, which will speed up generation of the move command by the manipulator/robotic prosthesis control software while also improving the accuracy of its positioning. In the context of application of the suggested algorithm, CVS will be used only to control the correctness of execution of an action by the manipulator, which greatly simplifies the algorithmic part, reduces the computational load and time to generate a command. This approach will increase the safety of medical rehabilitation robotic devices for which the key performance criteria are speed of reaction and accuracy of execution of actions.

References

1. Lassig R, Lorenz M, Sissimators E, Wicker I, Buchner T. Robotics Outlook 2030: How intelligence and mobility will shape the future. 2021. The Boston Consulting Group.
2. Joseph F. Robotics in Service. Engelberger, 1989; 248 p.
3. Alemzadeh H, Raman J, Leveson N, Kalbarczyk Z, Iyer RK. Adverse Events in Robotic Surgery: A Retrospective Study of 14 Years of FDA Data. PLOS ONE. 2016; 11 (4): e0151470. DOI: 10.1371/journal.pone.0151470.
4. Taylor GR, Kleeman L. Grasping unknown objects with a humanoid robot. 2002.
5. Kragic D, Christensen HA. Survey on Visual Servoing for Manipulation. Comput Vis Act Percept Lab. 2002; 15 p.
6. Dong G, Zhu ZH. Position-based visual servo control of autonomous robotic manipulators. Acta Astronautica. 2015; 115: 291–302. DOI: 10.1016/j.actaastro.2015.05.036.
7. Vahrenkamp N, Wieland S, Azad P, Gonzalez D, Asfour T, Dillmann R. Visual servoing for humanoid grasping and manipulation tasks. Humanoids 2008 — 8th IEEE-RAS International Conference on Humanoid Robots. 2000. DOI: 10.1109/ichr.2008.4755985.
8. Chaumette F, Hutchinson S. Visual servo control, Part I: Basic approaches. IEEE Robotics and Automation Magazine. 2006; 13 (4): 82–90.
9. Espiau B, Chaumette F, Rives P. A new approach to visual servoing in robotics. IEEE Transactions on Robotics and Automation. 1992; 8 (3): 313–26. DOI: 10.1109/70.143350.
10. Chaumette F. Potential problems of stability and convergence in image-based and position-based visual servoing. Lecture Notes in Control and Information Sciences. 66–78. DOI: 10.1007/bfb0109663.
11. Michel H, Rives P. Singularities in the determination of the situation of a robot efector from the perspective view of 3 points. Technical Report 1850, INRIA Research, 1993.
12. Chaumette F. 2004. Image Moments: A General and Useful Set of Features for Visual Servoing. IEEE Transactions on Robotics. 20 (4): 713–23. DOI: 10.1109/tro.2004.829463.
13. Tahri O, Chaumette F. Point-based and region-based image moments for visual servoing of planar objects. IEEE Transactions on Robotics. 2005; 21 (6): 1116–27. DOI: 10.1109/TRO.2005.853500.
14. Tahri O, Mezouar Y, Chaumette F, Corke P. Decoupled Image-Based Visual Servoing for Cameras Obeying the Unified Projection Model. Robotics, IEEE Transactions on. 2010; 26: 684–97. DOI: 10.1109/TRO.2010.2051593.
15. Dementhon DF, Davis LS. Model-based object pose in 25 lines of code. International Journal of Computer Vision. 1995; 15 (1–2): 123–41. DOI: 10.1007/bf01450852.
16. Lowe DG. Three-dimensional object recognition from single two-dimensional images. Artif Intell. 1987; 31: 355–95.
17. Malis E, Chaumette F, Boudet S. 2-1/2D visual servoing. IEEE Transactions on Robotics and Automation. 1999; 15 (2): 238–50.
18. Wilson WJ, Hulls CCW, Bell GS. Relative end-effector control using Cartesian position based visual servoing. IEEE Transactions on Robotics and Automation. 1996; 12 (5): 684–96. DOI: 10.1109/70.538974.
19. Corke PI. Visual control of robot manipulators — a review. Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback. 1993; pp. 1–31.
20. Corke PI, Hutchinson SA. A new partitioned approach to image-based visual servo control. IEEE Transactions on Robotics and Automation. 2001; 17 (4): 507–15. DOI: 10.1109/70.954764.
21. Deguchi K. Optimal motion control for image-based visual servoing by decoupling translation and rotation. Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. #. 98CH36190). DOI: 10.1109/iros.1998.727274.
22. Gans NR, Hutchinson SA. Stable Visual Servoing Through Hybrid Switched-System Control. IEEE Transactions on Robotics. 2007; 23 (3): 530–40. DOI: 10.1109/tro.2007.895067.
23. Kermorgant O, Chaumette F. Combining IBVS and PBVS to ensure the visibility constraint. 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2011. DOI: 10.1109/iros.2011.6048254.
24. Pang Y, Huang Q, Jia D, Tian Y, Gao J, Zhang W. Object manipulation of a humanoid robot based on visual Servoing. 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2007. DOI: 10.1109/iros.2007.4399445.
25. Moughlbay AA, Cervera E, Martinet P. Real-time model based visual servoing tasks on a humanoid robot. Intelligent Autonomous Systems 12. Berlin, Heidelberg: Springer, 2013; c. 321–333.
26. Agravante DJ, Cherubini A, Bussy A, Kheddar A. Human-humanoid joint haptic table carrying task with height stabilization using vision. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2013. DOI: 10.1109/iros.2013.6697019.
27. Hogan N. Impedance Control: An Approach to Manipulation. American Control Conference. 1984; pp. 304–313. DOI: 10.23919/ACC.1984.4788393.
28. Aristidou A, Lasenby J. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. Graphical Models. 2011; 73 (5): 243–60. DOI: 10.1016/j.gmod.2011.05.003.

Литература

1. Lassig R, Lorenz M, Sissimators E, Wicker I, Buchner T. Robotics Outlook 2030: How intelligence and mobility will shape the future. 2021. The Boston Consulting Group.
2. Joseph F. Robotics in Service. Engelberger, 1989; 248 p.
3. Alemzadeh H, Raman J, Leveson N, Kalbarczyk Z, Iyer RK. Adverse Events in Robotic Surgery: A Retrospective Study of 14 Years of FDA Data. PLOS ONE. 2016; 11 (4): e0151470. DOI: 10.1371/journal.pone.0151470.
4. Taylor GR, Kleeman L. Grasping unknown objects with a humanoid robot. 2002.
5. Kragic D, Christensen HA. Survey on Visual Servoing for Manipulation. Comput Vis Act Percept Lab. 2002; 15 p.
6. Dong G, Zhu ZH. Position-based visual servo control of autonomous robotic manipulators. Acta Astronautica. 2015; 115: 291–302. DOI: 10.1016/j.actaastro.2015.05.036.
7. Vahrenkamp N, Wieland S, Azad P, Gonzalez D, Asfour T, Dillmann R. Visual servoing for humanoid grasping and manipulation tasks. Humanoids 2008 — 8th IEEE-RAS International Conference on Humanoid Robots. 2000. DOI: 10.1109/ichr.2008.4755985.
8. Chaumette F, Hutchinson S. Visual servo control, Part I: Basic approaches. IEEE Robotics and Automation Magazine. 2006; 13 (4): 82–90.

9. Espiau B, Chaumette F, Rives P. A new approach to visual servoing in robotics. IEEE Transactions on Robotics and Automation. 1992; 8 (3): 313–26. DOI: 10.1109/70.143350.

10. Chaumette F. Potential problems of stability and convergence in image-based and position-based visual servoing. Lecture Notes in Control and Information Sciences. 66–78. DOI: 10.1007/bfb0109663.

11. Michel H, Rives P. Singularities in the determination of the situation of a robot efector from the perspective view of 3 points. Technical Report 1850, INRIA Research, 1993.

12. Chaumette F. 2004. Image Moments: A General and Useful Set of Features for Visual Servoing. IEEE Transactions on Robotics. 20 (4): 713–23. DOI: 10.1109/tro.2004.829463.

13. Tahri O, Chaumette F. Point-based and region-based image moments for visual servoing of planar objects. IEEE Transactions on Robotics. 2005; 21 (6): 1116–27. DOI: 10.1109/TRO.2005.853500.

14. Tahri O, Mezouar Y, Chaumette F, Corke P. Decoupled Image-Based Visual Servoing for Cameras Obeying the Unified Projection Model. Robotics, IEEE Transactions on. 2010; 26: 684–97. DOI: 10.1109/TRO.2010.2051593.

15. Dementhon DF, Davis LS. Model-based object pose in 25 lines of code. International Journal of Computer Vision. 1995; 15 (1–2): 123–41. DOI: 10.1007/bf01450852.

16. Lowe DG. Three-dimensional object recognition from single two-dimensional images. Artif Intell. 1987; 31: 355–95.

17. Malis E, Chaumette F, Boudet S. 2-1/2D visual servoing. IEEE Transactions on Robotics and Automation. 1999; 15 (2): 238–50.

18. Wilson WJ, Hulls CCW, Bell GS. Relative end-effector control using Cartesian position based visual servoing. IEEE Transactions on Robotics and Automation. 1996; 12 (5): 684–96. DOI: 10.1109/70.538974.

19. Corke PI. Visual control of robot manipulators — a review. Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback. 1993; pp. 1–31.

20. Corke PI, Hutchinson SA. A new partitioned approach to image-based visual servo control. IEEE Transactions on Robotics and Automation. 2001; 17 (4): 507–15. DOI: 10.1109/70.954764.

21. Deguchi K. Optimal motion control for image-based visual servoing by decoupling translation and rotation. Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. #. 98CH36190). DOI: 10.1109/iros.1998.727274.

22. Gans NR, Hutchinson SA. Stable Visual Servoing Through Hybrid Switched-System Control. IEEE Transactions on Robotics. 2007; 23 (3): 530–40. DOI: 10.1109/tro.2007.895067.

23. Kermorgant O, Chaumette F. Combining IBVS and PBVS to ensure the visibility constraint. 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2011. DOI: 10.1109/iros.2011.6048254.

24. Pang Y, Huang Q, Jia D, Tian Y, Gao J, Zhang W. Object manipulation of a humanoid robot based on visual Servoing. 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2007. DOI: 10.1109/iros.2007.4399445.

25. Moughlbay AA, Cervera E, Martinet P. Real-time model based visual servoing tasks on a humanoid robot. Intelligent Autonomous Systems 12. Berlin, Heidelberg: Springer, 2013; c. 321–333.

26. Agravante DJ, Cherubini A, Bussy A, Kheddar A. Human-humanoid joint haptic table carrying task with height stabilization using vision. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2013. DOI: 10.1109/iros.2013.6697019.

27. Hogan N. Impedance Control: An Approach to Manipulation. American Control Conference. 1984; pp. 304–313. DOI: 10.23919/ACC.1984.4788393.

28. Aristidou A, Lasenby J. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. Graphical Models. 2011; 73 (5): 243–60. DOI: 10.1016/j.gmod.2011.05.003.