

Low Cost Robotic Arm for Object Grasping Applications

Sanja Golomeova

Faculty of Computer Science, University "Goce
Delchev" - Shtip
"Krste Misirkov" No.10/A, 2000 Stip, Macedonia

Saso Koceski

Faculty of Computer Science, University "Goce
Delchev" - Shtip
"Krste Misirkov" No.10/A, 2000 Stip, Macedonia

ABSTRACT

Initially, the emphasis of the robotic arm is placed on grasping and positioning objects in space. These types of robotic arms have a stake in planetary and underwater research, robotic surgery, systems for disabled people, households and so on. The strategy for grasping and positioning objects is based on recording environment with a web camera. Strategic decisions affect the orientation of the hand and the way of opening the gripper. At the same time, opening the gripper should match the shape and size of objects being used. The creation of a system based on this strategy, which then can be practically applied while being inexpensive and simple to implement, is the main goal of this paper.

Keywords

Robotic arm, manipulator, grasping objects, segmentation, visual servoing

1. INTRODUCTION

Robotic arms are electro-mechanical systems also known as anthropomorphic hands because of their mechanical structure and functions. The control of their functions is performed by computer interface and, for developing rapport with environment, they use many sensors and many different applications such as: applications for grasping [1], positioning [2], moving objects, as well as for welding [3], painting, assembly [4] and transportation.

These applications are not integrated and used only in industrial factories; they are also used in our daily lives [5], in medicine [6] and in space. Today, more and more of these applications are used in surgery.

Robotic arm for object grasping applications or any other type of applications must be composed of sensors, motors and end-effector. Sensors are used for collecting information about the environment. They solve problems of tracking and determine location, path and characteristic of definite objects. Cameras, microphones, ultrasounds, 3D scanners [7] and other devices can be used as sensors. The motors are the place where the motion of the arm occurs and the end-effector is the last part which grasps and moves objects.

Robotic arms which use a camera as a sensor for determining the space for positioning objects and the algorithm for tracking the objects are part of the relatively new branch of robotic science called vision-based robot control or visual servoing. These robotic arms are classified in 2 groups: by camera position and tracking algorithm. By camera position, there are two main types of arms: eye-in-hand (where the camera is attached to the robotic arm end-effector) and eye-to-hand (where the camera is fixed somewhere in the environment around the robot arm manipulator). By tracking algorithm there also two types of arms: real time robotic arms tracking or robotic arms with active tracking (here tracking and determining the motion and location of objects are via

video camera) and robotic arms with passive tracking (here robotic arms store information in devices and by analyzing this information on a computer it determines the location of objects) [8-9].

The tracking algorithm provides future positions of multiple moving objects based on their old positions by image processing from the camera. From processing the image, the algorithm needs to take out objects of interest with the method of image segmentation and then manage with them. Therefore, these systems are also called 'look-then-move' systems and here the accuracy of the systems depends directly on the accuracy of the visual sensor and the robotic arm.

This paper is focused on eye-to-hand robotic arm like visual servoing system with active tracking. This system uses a stationed camera on the work environment of robotic arm, USB port for transferring information to the computer, USBOR servo controller that controls servo motors of the robotic arm and via forward kinematic is defining the position of the moving objects. The system has been tested at a fixed speed and has 6 degrees of freedom movement of the robotic arm [10].

2. DESCRIPTION OF THE DEVELOPED ROBOT ARM

Robix Rascal robotic arm has 6 degrees of freedom (DOF) which makes it similar to a human hand. The arm can make three different ways of rotating solid and three different ways of translations. All movements together are creating the 6 DOF robotic arm. Starting from the fact that the degrees of freedom of movement are determined by the number of joints, it is more than clear that it is composed of 6 joints. All joints are of the revolute type. In the place of joints there are servo motors that have the same role and perform the same function. Their function is to enable movements of the hand and to insert special mobility in the arm flexibility. These joints (servo motors) establish the relation to each other with links. Links are connected as one chain (link-joint-link-joint) and, because of this, the arm belongs to the group of serial chain manipulators. The chain of links and joints is finalized with an end-effector, which has the main function of the hand and performs the task which is the goal of the robotic arm. The end effector has two nippers (fingers) that have the task of grasping objects and opening and closing fingers, alternately. Its other name is gripper, due to its ability. Here, the end-effector is free, it means that it can move freely in space, from up to down, from left to right, from open to close and backward. Because of their complexity, in all robotic arms the end effectors are considered as separate subsystems, which analyze and process separately from the robotic arm. And their degrees of freedom are defined separately.

To enable the end effector for grasping and positioning objects, it is necessary to collect information about these objects and map where the robotic arm should position them.

Sensors are used as a solution for this problem of object tracking. In Rascal robotic arm we use a web camera as a sensor. The camera has a fixed position; its coordinates are fixed in the environment up to the map where the robotic arm needs to position the objects. The image processed from the camera gives all the necessary information processed in real-time, so this tells us that it is active object tracking. [11]

The active object tracking robotic arm performs its function in several steps: first, it detects target objects and positions the map; second, it extracts information from the picture (the initial position of the object, the coordinates of the target point on the map), and the third step is to follow that information, grasp the object and position it in the target point.

For an accurate and precise realization of the process, the robotic arm is controlled by a computer. It is related to computer via USB port and all the information it receives from there. Robix Rascal robotic arm is commanded via Usbor servo controller, which generates pulses for move servo motors of the robotic arm. The tracking algorithm is responsible for the accuracy and precision of the robotic arm. The authenticity of the robotic arm depends on its authenticity (Fig. 1).

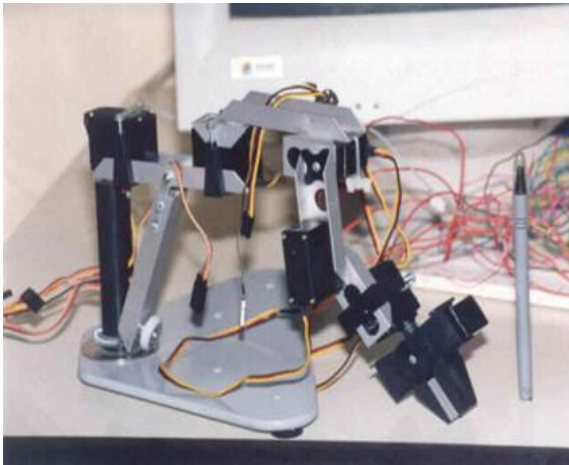


Fig 1: The Robix Rascal robotic arm with all hardware parts

2.1 Forward and Inverse kinematics

This robotic arm consists of 6 joints (servo motors). All joints are revolute and can move forward, backward, up, down and rotate (left and right).

Just like in the human arm, the robotic arm with all joints is divided into 3 parts: shoulder, elbow and wrist. Every joint has three axes of movement: x, y and z, which are also known as coordinate frames. All joints are connected with links.

Fig. 2 provides different coordinate frames and links' lengths. In Table 1 the numerical values of the lengths of all the 6 links from "wrist - wrist" are given.

The Forward kinematics model solves the problem of the position and orientation of the end-effector of the robot, if we know the angle of rotation and the axis. The best methodology for describing the forward kinematics of a robotic hand is Denavit - Hartenberg (DH) convention and methodology. This convention standardized coordinate frames to each joint for representing the relationship between them in the robotic arm.

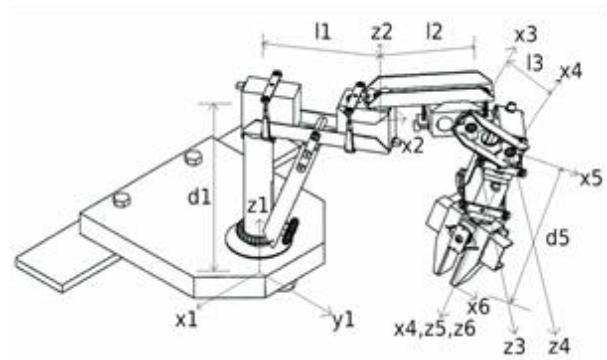


Fig 2: Different coordinate frames and links lengths

Table 1. Numerical values of the lengths of links on the robotic arm

Markings on the links	Description	Dimensions (cm/holes)
b_s	basis - shoulder	12cm
s_e1	shoulder - elbow1	10cm/5
e1_e2	elbow1 – elbow2	10cm/5
e2_e3	elbow2 – elbow3	12cm/6
e3_e4	elbow3 – elbow4	12cm/6
e4_w	elbow4 – wrist	6cm/3

The Convention is composed of four D-H parameters [12-15]:

- θ_i is the joint angle
- d_i is the joint offset
- a_i is the link length
- α_i is the twist angle

Joint 1 (shoulder J1) is the Base of arm and it rotates for angle θ_1 around z_1 axis. Joint 2 (J2) is identified as the J1, with the difference that it rotates for angle θ_2 around z_2 axis. Joint 3(J3) and Joint 4(J4) translates for d_1 units in x_3y_3 and x_4y_4 plane. Joint 5(J5) is a wrist joint, it translates for d_2 unites in x_5y_5 plane. And the last joint 6 is the gripper joint, it gives the grip's roll.

For Robix Rascal robotic hand D-H parameters for each joint with their values are given in Table 2.

Following the DH methodology, the general transformation matrices, which express the position and orientation of the gripper with respect to the arm's base, are computed with Equations (1).

Table 2. D-H parameters of 6 joints Robix Rascal robotic arm

Joints	d(mm)	θ (deg)	a(mm)	α (deg)
J1	0	90	120	0
J2	20	90	100	0
J3	20	90	100	0

J4	20	90	120	45
J5	20	90	120	0
J6	20	90	60	45

$$R_z(\theta_n) = \begin{bmatrix} \cos 90^\circ & -\sin 90^\circ & 0 & 0 \\ \sin 90^\circ & \cos 90^\circ & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_z(d_n) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 20 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$R_x(\alpha_n) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 45^\circ & -\sin 45^\circ & 0 \\ 0 & \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

When we have the values of position and orientation of end-effector but we would like to calculate joint angles, we need to use inverse kinematics. This kinematics is more complex and difficult for solving in contrast to forward kinematics. And because of this problems and errors in the calculations often occur. Inverse kinematics (changes in the angles of joints is a function of the particular position of the end effector) is obtained by inverse Jacobian method. [13]

The mathematical representation of the inverse kinematics technique is defined in Equation (2). \vec{Z}_i is the position on the end effector and it has the value $\{x, y, z\}$, $\Delta\vec{\theta}$ is the angle of rotation of z-axis. [13]

$$\Delta\vec{\theta} = J^{-1}(\vec{\theta}) \cdot \Delta\vec{Z} \quad (2)$$

3. GRASPING APPLICATION

Grasping application is based on the visual servoing technique. This means that whether the robotic arm will grasp the objects depends on the feedback information extracted from the camera. Correct programming and work of the camera are some of the main elements of good application.

The application runs in four main steps: the arm opens fingers for grasp, then tracks the object, swoops the object and positions it to the map. In order to follow these steps the robotic arm needs to know the coordinate frames of objects and the map. Coordinate frames are received from the image reproduced from the camera. With the method of segmentation (thresholding) of the image, the robotic arm gets the coordinate frames and recognizes the positions (Fig. 3).

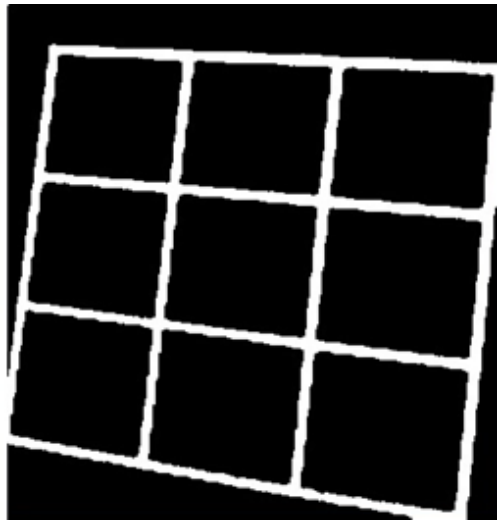
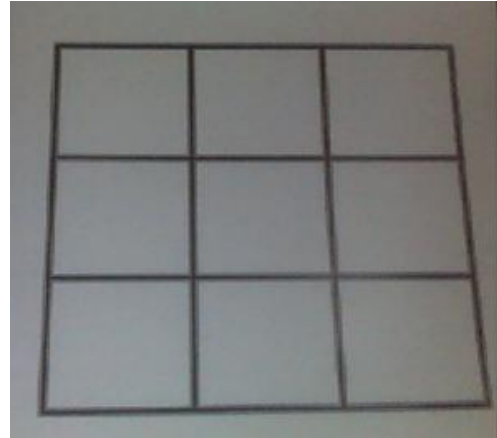


Fig 3: Upper image is the original image of the map where the objects should be placed and bottom image is the threshold of the original image

The algorithm provides the necessary transformations of the robotic manipulator, rotation and translation, to enable manipulator motion. Executing three series of transformations: one for grasping objects, another for their positioning, and the third transformation is the inverse transformation for the return of the robotic arm in its original position.

Application is fast, simple and comprehensible. It is fast because the robotic arm grasps all objects with fixed speed. The simplicity of the application is explained by the process of run. The whole process is: when the application is started from the computer, the robotic manipulator opens its nippers and makes some transformations for tracking the objects; at that moment it is seeking to reach the coordinate frames (x, y, z) of the target object and when it positions its gripper in that coordinate frames, it swoops the target object. Next, it recognizes the image and the free points on it where it can position the object, picks one and moves towards it (again with some transformations). When the robotic manipulator comes to the point of positioning, it places the object in it and returns to its original position (Fig. 4). The process is iterative, depending on the number of objects.

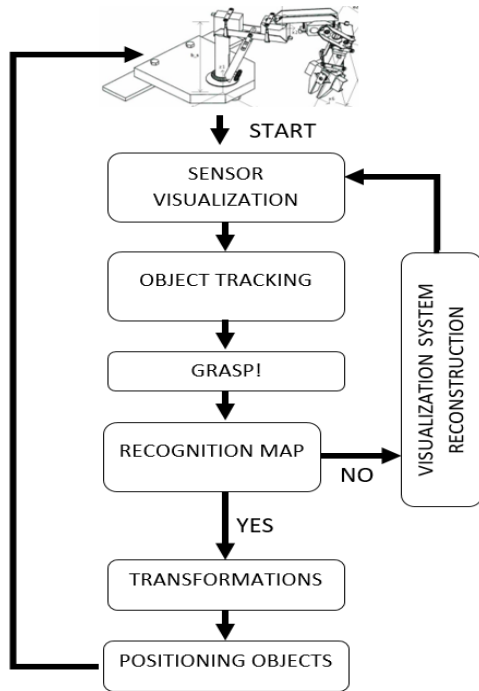


Fig 4: Flowchart of the grasping and positioning application

4. EVALUATION AND CONCLUSION

The algorithm and statistics from the robot arm evaluation that will be shown below in this chapter are steps in achieving one of the goals mentioned in the Introduction chapter. The robotic arm that we use is Rascal's robotic arm, made by Robix. This robotic arm is fitted with a 6 DOF grip.

We experimented and tested the robot in several steps:

- First, it was done by setting up a webcam to monitor the objects.
- Second, testing was done with the same objects (cubes), but in different positions and at different speeds of operation of the robotic arm (Table 3).

Based on these data, we have seen the probability of an error by the robot. The potential deviations of the robotic arm from the object are given in cm.

The final result shows that as the speed of the robotic arm increases and the distance of the object from the position of the robotic arm also increases, the possibility of the probable error also increases. And, if this deviation is drastically expressed, then a robust deviation of the robotic arm from the object can occur and thus the precision of the robot can be compromised.

Table 3. Trial from 6 cubes on different places and different velocity of the robotic arm

Objects	Velocity of the robotic arm	Distance between the robotic arm and objects (m)	Grasping (%)	Deviation (m)
1	100	0,2	100%	0
2	50	0,5	100%	0
3	40	0,1	100%	0
4	200	0,25	80%	0,02m
5	200	0,42	60%	0,04m
6	80	0,13	100%	0

This deviation is shown in Figure 5. For the initial value (0,0) of the robotic arm and for the same time of the grasping for all objects, the velocity and acceleration have deviations.

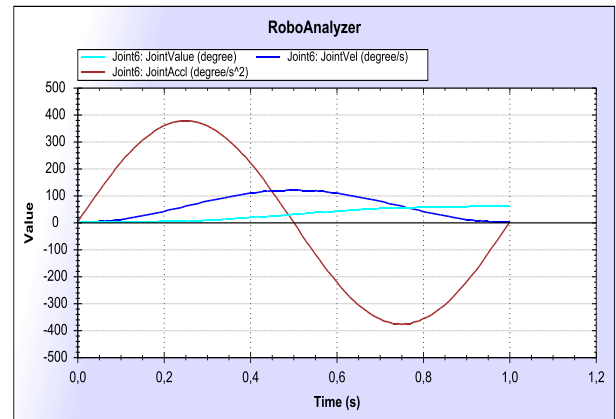


Fig 5: Deviation between time and initial value of the robotic arm

That is because all objects are at different distances from the robotic arm and the time of grasping for all of them is the same. Thus the robotic arm grasps with small velocity/acceleration the objects that are near and the arm is precise in grasping the near objects. And another explanation is that the arm, when it is more distant from the objects, will have greater velocity/acceleration for the same time and the deviation in the result will be bigger and the result incorrect. The obtained results are encouraging and our future research will be oriented towards development of a multiple robot arm system for collaborative working.

5. REFERENCES

- [1] Ciocarlie, M., Hsiao, K., Jones, E.G., Chitta, S., Rusu, R.B. and Şucan, I.A., 2014. Towards reliable grasping and manipulation in household environments. In Experimental Robotics (pp. 241-252). Springer Berlin Heidelberg.

- [2] Katzschmann, R.K., Marchese, A.D. and Rus, D., 2015. Autonomous object manipulation using a soft planar grasping manipulator. *Soft robotics*, 2(4), pp.155-164.
- [3] Liu, Y. and Zhang, Y., 2015. Toward welding robot with human knowledge: A remotely-controlled approach. *IEEE Transactions on Automation Science and Engineering*, 12(2), pp.769-774.
- [4] Chen, Y., Xie, F., Liu, X. and Zhou, Y., 2014. Error modeling and sensitivity analysis of a parallel robot with SCARA (selective compliance assembly robot arm) motions. *Chinese Journal of Mechanical Engineering*, 27(4), pp.693-702.
- [5] Koceski, S., Koceska, N. and Kocev, I., 2012. Design and evaluation of cell phone pointing interface for robot control. *International Journal of Advanced Robotic Systems*, 9(4), p.135.
- [6] Koceski, S. and Koceska, N., 2016. Evaluation of an assistive telepresence robot for elderly healthcare. *Journal of medical systems*, 40(5), p.121.
- [7] Koceski, S., Koceska, N., Zobel, P.B. and Durante, F., 2009, June. Characterization and modeling of a 3D scanner for mobile robot navigation. In *Control and Automation, 2009. MED'09. 17th Mediterranean Conference on* (pp. 79-84). Ieee.
- [8] Xiao, X., Huang, Z., Rube, M.A. and Melzer, A., 2017. Investigation of active tracking for robotic arm assisted magnetic resonance guided focused ultrasound ablation. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 13(3).
- [9] Wang Y., Ewert D., Vossen R., Jeschke S. A Visual Servoing System for Interactive Human-Robot Object Transfer. *Journal of Automation and Control Engineering* Vol. 3, No. 4, 277-278, 281-282 (2015).
- [10] Mourya R., Shelke A., Satpute S., Kakade S., Botre M.: Design and Implementation of Pick and Place Robotic Arm. *International Journal of Recent Research in Civil and Mechanical Engineering (IJRRCME)*, 233-234. (2015)
- [11] Hu, A.P., Thomas, S., Jennings, B., McMurray, G. and Rains, G.C., 2014. Development and Testing of a Low-Cost Multi-Function Robotic Arm for Agriculture. In 2014 Montreal, Quebec Canada July 13–July 16, 2014 (p. 1). American Society of Agricultural and Biological Engineers.
- [12] Mulande S. N., Sharma K. K. and Huidrom R.: Development of an algorithm for analysis of workspace of the Robix robot manipulator using Matlab. *Journal of Material Science and Mechanical Engineering (JMSME)*, Volume 2, Number 3, 231. (April-June 2015)
- [13] Goto S.: *Robot Arms*. Saga University, Japan, 24-35. (2011)
- [14] Corke P.: Denavit-Hartenberg notation for common robots. (March 2014)
- [15] Harpreet Singh E., Naveen Dhillon D., and Imran Ansari E.: Forward and inverse Kinematics Solution for Six DOF with the help of Robotics tool box in matlab. *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, Volume 4, Issue 3, 18. (2015)
- [16] Mourya R., Shelke A., Satpute S., Kakade S., Botre M.: Design and Implementation of Pick and Place Robotic Arm. *International Journal of Recent Research in Civil and Mechanical Engineering (IJRRCME)*, 233-234. (2015).