Modeling and testing of control system for an underwater dual-arm robot

V Kramar*, A Kabanov, O Kramar and A Putin
Sevastopol State University, 299053, Universitetskaya, 33, Sevastopol, Russia

* kramarv@mail.ru

Abstract. The article exam the approach to testing the software of the control system for an underwater dual-arm robot. The proposed approaches are related both to the modeling of robot computer models in Matlab, to testing the control system software by a dual-arm robot simulator and to testing software by the real robot. The proposed testing approach made it possible to verify the correctness of the developed control system software by evaluating parameters such as accuracy and repeatability. An example of a test procedure using the robot simulator is given.

1. Introduction
The development of the robots is a time-consuming process which consist of many stages. The testing process is one of the important stages, along with the development of the robot design and control system. There are various types of testing, such as performance testing, reliability testing and others. The reliability testing is the main in the testing process the robot control system [1]. The purpose of the reliability testing is to obtain information about errors, in particular, errors resulting from the operation of the control system [2]. One of the types of reliability testing is acceptance testing [2]. The form of organization of acceptance testing can be verification of achievement of design goals. A key role in the control system testing process for a dual-arm robot is played by parameters such as accuracy and repeatability [1]. The accuracy determines the difference between the target and the real pose of the end-effector. The repeatability determines the proximity of the coincidence of the positions of the robot-manipulators during repeated movements under the same external conditions. These parameters can be applicable to testing the movement of the robot arm along a given path.

The test conditions for the same robot-manipulator and the scenario should be similar. The external conditions of the tested robot-manipulator must be constant, and the robot-manipulator must be in a stable position. [1].

A number of test methods for evaluating performance are known. This is the IPA-Stuttgart method, the Ford method, the test-robot program method, the National Bureau of Standards method, and the Chesapeake laser systems method [3]. Briefly describe these methods. The IPA-Stuttgart method [1,4] describes a procedure for measuring static positioning errors and dynamic path parameters. Moreover, the method requires precise coordination between the robot and the measuring equipment [4]. The Ford method consists in the fact that the robot is subjected to a cyclic test, which consists in moving all axes from one end of the movement to the other at full nominal payload and at a speed of 40 hours continuously. The purpose of this method is to establish the reliability of the system. The idea of the test-robot program method is to measure positioning errors and orientations obtained during positioning. The test-robot program method has a number of functions consisting in: the ability to perform error
analysis of the position and orientation of the robot arm; the ability to be used as a modeling tool for evaluating alternative solutions; the ability to evaluate perpendicularity errors of the tested robot; the ability to handle input/output errors on disk and edit input data at run time; the ability to save the results in a user-specified text file on disk [5]. The National Bureau of Standards method and Chesapeake laser systems method are based on the use of laser measuring technology [1].

The purpose of this article is to describe the testing process of developed algorithms and software for the underwater dual-arm robot control system. The rest of the paper is as follows. Section 2 describes the object of the testing. The computer model of the dual-arm robot is shown in Section 3. The conditions, procedure, types, stages and test results discussed and the conclusions are given in sections 4, 5.

2. The dual-arm robot as the object of the testing
An underwater dual-arm robot (Fig.1) is described as a test robot.

![Figure 1. The design of the considered dual-arm robot.](image)

According to the assignment for the development of an anthropomorphic robot arm control system, we will choose the ideas of the test-robot program method. We will take into account that the test methods of the robot arm control program include: the modeling and the testing both on a computer model and on a real robot. The modeling is performed according to the constructed mathematical models in the Matlab. The modeling allows us to establish that the control system software correctly implements this stage of verification. The testing through a computer model is performed using a simulator corresponding to the considered dual-arm robot. The final testing is performed on the real robot. For each of the verification steps, the set of test data for modeling coincides with the set of test data used for the real robot testing.

3. The computer model of the dual-arm robot
The basis for testing a software for the dual-arm robot control system is the development of its computer model. A model for describing dual-arm robot motion based on the application of the Denavit-Hartenberg approach was chosen as a computer model. A detailed description of the model is given in [6]. We indicate the Denavit-Hartenberg parameters for the dual-arm robot in Table 1.

| Parameter | Description |
|-----------|-------------|
| $A_i$ | the shortest distance of adjacent links between the axes $Z_{i-1}$ and $Z_i$, measured along their common normal; |
| $\alpha_i$ | the angle between the axes of the joints, measured in a plane perpendicular to their common normal; |
| $d_i$ | the distance between adjacent links, counted along the joint axis of the previous link; |
| $\theta_i$ | the angle between the normal, measured in a plane perpendicular to the axis of the previous link. |

In table 1: $A_i$ - the shortest distance of adjacent links between the axes $Z_{i-1}$ and $Z_i$, measured along their common normal; $\alpha_i$ - the angle between the axes of the joints, measured in a plane perpendicular to their common normal; $d_i$ - the distance between adjacent links, counted along the joint axis of the previous link; $\theta_i$ - the angle between the normal, measured in a plane perpendicular to the axis of the previous link.
Also, in order to build an adequate computer model, it is necessary to take into account the design constraints established at the angles of the robot arm links. The indicated restrictions are given in table 2. This approach to development of the computer model seems adequate, based on the fact that the control of the robot arms consists in calculating and indicating to the control system the values of the rotation angles and the rotation speed of the motors of the robot links.

Table 1. Denavit-Hartenberg parameters table for the robot arms

| Joints of the right robot arm | \( A_i \) | \( \alpha_i \) | \( d_i \) | \( \theta_i \) | Joints of the left robot arm | \( A_i \) | \( \alpha_i \) | \( d_i \) | \( \theta_i \) |
|-----------------------------|---------|---------|---------|---------|-----------------------------|---------|---------|---------|---------|
| 1                           | -0.2    | \( \pi/2 \) | -0.2    | \( \theta_i \) | 1                           | -0.2    | \( \pi/2 \) | 0.2    | \( \theta_i \) |
| 2                           | 0       | \( \pi/2 \) | 0       | \( \theta_i \) | 2                           | 0       | \( \pi/2 \) | 0       | \( \theta_i \) |
| 3                           | 0       | \( \pi/2 \) | 0.47    | \( \theta_i \) | 3                           | 0       | \( \pi/2 \) | 0.47   | \( \theta_i \) |
| 4                           | 0       | \( \pi/2 \) | 0       | \( \theta_i \) | 4                           | 0       | \( \pi/2 \) | 0       | \( \theta_i \) |
| 5                           | 0       | \( \pi/2 \) | 0.2     | \( \theta_i \) | 5                           | 0       | \( \pi/2 \) | 0.2    | \( \theta_i \) |
| 6                           | 0.2     | 0       | 0       | \( \theta_i \) | 6                           | 0.5     | 0       | 0       | \( \theta_i \) |

Table 2. Limitations on the angles of the robot arms

| Link of the right robot arm | Range | Link of the left robot arm | Range |
|-----------------------------|-------|-----------------------------|-------|
| 1                           | \(-90^\circ \leq \theta \leq 90^\circ\) | 1     | \(-90^\circ \leq \theta \leq 90^\circ\) |
| 2                           | \(0^\circ \leq \theta \leq 120^\circ\) | 2     | \(-120^\circ \leq \theta \leq 0^\circ\) |
| 3                           | \(-90^\circ \leq \theta \leq +90^\circ\) | 3     | \(-90^\circ \leq \theta \leq +90^\circ\) |
| 4                           | \(-110^\circ \leq \theta \leq 30^\circ\) | 4     | \(-110^\circ \leq \theta \leq 30^\circ\) |
| 5                           | \(-80^\circ \leq \theta \leq 80^\circ\) | 5     | \(-80^\circ \leq \theta \leq 80^\circ\) |
| 6                           | \(-90^\circ \leq \theta \leq 90^\circ\) | 6     | \(-90^\circ \leq \theta \leq 90^\circ\) |

3.1. The Matlab computer model

Based on the fact that we chose modeling as the first step, we present a computer model of the robot in Matlab. To create a computer model in Matlab, we use the data from table 1, table 2 and the syntax used in Toolbox RTB 10 [7].

The computer model code for Matlab is shown in Fig. 2.

```matlab
initialPose_left = deg2rad([0 0 0 0 0]);
L(1) = Revolute('d', 0.2, 'alpha', pi/2, 'a', -0.2, 'offset',
deg2rad(0), 'qlim', initialPose_left(1)+deg2rad([-89 9]));
L(2) = Revolute('d', 0, 'alpha', pi/2, 'a', 0, 'offset',
deg2rad(-90), 'qlim', initialPose_left(2)+deg2rad([-120 0]));
L(3) = Revolute('d', 0.47, 'alpha', pi/2, 'a', 0, 'offset',
deg2rad(-90), 'qlim', initialPose_left(3)+deg2rad([-90 +90]));
L(4) = Revolute('d', 0, 'alpha', pi/2, 'a', 0, 'offset',
deg2rad(-90), 'qlim', initialPose_left(4)+deg2rad([-110 30]));
L(5) = Revolute('d', 0.2, 'alpha', pi/2, 'a', 0, 'offset',
0, 'qlim', initialPose_left(5)+deg2rad([-80 80]));
L(6) = Revolute('d', 0, 'alpha', 0, 'a', 0.5, 'offset',
deg2rad(90), 'qlim', initialPose_left(6)+deg2rad([-30 90]));
baseL = [0 0 -1 -0.2; 0 1 0 0; 1 0 0 0; 0 0 0 1];
```

Figure 2. The Matlab Script for the description of the left robot arm.
For the right robot arm we have a similar script, taking into account the data of tables 1 and 2. The graphical display of the robot computer model in Matlab is shown in Fig. 3.

![Figure 3. The graphical display of the computer model in Matlab.](image)

### 3.2. The robot simulator computer model

The robot simulator is “3D Simulator. The dual-arm robot for an underwater technical work” v.1.0. The software is designed to work with the experimental sample of the robot in the remote control mode by the operator by means of the copying device and an integrated 3D vision system. The computer model of the simulator is fully consistent with the robot. The main view of the simulator is shown in Fig. 4.

![Figure 4. The main view of the 3D robot simulator](image)
3.3. The computer model for robot control system

The computer model is implemented in a software that is a Windows application written in the C#. The program consists of two modules, this is the user interface and the kernel. The kernel is a dynamic library and consists of communication and data processing subsystems. The implemented computer model is based on the Denavit-Hartenberg representation and the parameters obtained from the 3D vision system, which determine the prediction of the self-collision of robot arms and the prediction of collision with the external objects.

All interaction of the subsystems is done through the kernel module using events and callback functions, so we get relevant and time-synchronized data from different sources.

3.4. The Modelling of developed control algorithms

According to the developed test procedure, the scope of tests and tasks is specified. These tasks are performed on the computer models in Matlab.

For example, by setting the target coordinates for bringing the robot arms and the rotation angles of the end-effector according to the information from the 3D computer vision system, when testing the movement of the robot arms:

- (-0.56, 1.174, 0.037); rpy ([[-60.24, -37.71, -179.66]]); left;
- (0.512, 0.622, -0.541); rpy ([88.06, 14.11, 11.69]); right

the joint angles are calculated, and therefore the position for the robot arms is calculated.

By running the script in Matlab, we get a graph of the position of the robot manipulators, confirming the correctness of the developed computer model (see. Fig. 5). Matlab script starts the number of times specified in the test procedure.

![Figure 5. The modelling in Matlab.](image-url)
4. The conditions, procedure, types, stages and test results of the developed algorithms and methods for the control system of the robot arms

After confirming the correct operation of the developed computer models based on the test methodology, the testing is performed.

The testing is carried out on the basis of conditions developed in the test program, test procedure, types and stages of testing. Table 3 shows the developed parameters for testing using a simulator.

| №  | Actions                                         | System behaviour                                                                 |
|----|------------------------------------------------|----------------------------------------------------------------------------------|
| 1  | Run robot simulator                             | See the procedure for starting the control system in the corresponding “User Guide”. The simulator has started. The robot is displayed properly. |
|    |                                                 | The software activates the kinematic model of the robot.                           |
|    |                                                 | The software reads the coordinates received from the 3D vision system to move the robot arms to the specified coordinates from a given file. |
|    |                                                 | The software solves the inverse kinematic problem by generating the rotation values of the motors of the links of the robot arm. |
|    |                                                 | A check was made for the possibility of arriving at the indicated coordinates with a minimum error, taking into account the necessary attitude of the end-effector. |
|    |                                                 | If the previous step is successful, the motor rotation angles and the motor rotation time are transmitted via TCP protocol to the simulator. |
|    |                                                 | In the simulator, robot arms come to the specified coordinates.                   |
| 2  | Launch robot arms control software              |                                                                                  |
|    |                                                 | The accuracy of positioning of the end-effector of the robot arm with the given coordinates is checked |
|    |                                                 | According to the current state of the engines, the direct kinematics problem is solved. |
|    |                                                 | The obtained coordinate value and the direction angle of the end-effector are checked with the given coordinates for the accuracy of the task. |

According to the developed test procedure, the scope of tests and tasks is specified. These tasks are performed on the simulator and on the robot. By setting the target coordinates for the manipulators and the rotation angles of the end-effector according to the 3D vision system, we will set the same manipulator position parameters as in subsection 3.4.

Starting the developed control system program, we will see the movements of the manipulators both on the simulator and on the robot. The program connections for the control system with the simulator and the robot and data transfer is provided by the TCP protocol.

If the manipulators came to a predetermined position, then testing was successful. The developed control program runs once specified in the test methodology. The result of the simulator for the given task is shown in Fig. 6.

The similar conditions, procedure, types and stages of testing are developed for testing a real robot (table 4).
Table 4. The conditions, procedure, types and the stages of testing the robot.

| №  | Actions                                | System behaviour                                                                 |
|----|----------------------------------------|-----------------------------------------------------------------------------------|
| 1  | Run the robot                          | See the procedure for starting the control system in the corresponding “User Guide”.<br>A check is made of the presence of the robot arms in the initial position.<br>The robot arms are brought into working position. |
| 2  | Launch the control system of the robot  | See the procedure for starting the control system in the corresponding “User Guide”.<br>The control system PC is turned on and working properly. |
| 3  | Launch robot arms control software      | The software activates the kinematic model of the robot arms.<br>The software reads the coordinates received from the 3D vision system to move the robot arms to the specified coordinates from a given file.<br>The software solves the inverse kinematic problem by generating the rotation values of the engines of the links of the robot arm.<br>A check was made for the possibility of arriving at the indicated coordinates with a minimum error, taking into account the necessary turn of the end-effector.<br>If the previous point is successful, the engine rotation angles and the engine rotation time are transmitted via TCP protocol to the robot.<br>The control signals corresponding to the rotation angles are transmitted to the manipulator link engines.<br>The robot arms come to the specified coordinates.<br>The rotation angles of the engines are read and compared with the calculated ones.<br>If necessary, an automatic adjustment of the rotation angles to the specified ones is carried out. |
| 4  | The accuracy of positioning of the end-effector of the robot arm with the given coordinates is checked | According to the current state of the engines, the direct kinematics problem is solved.<br>The obtained coordinate value and the direction angle of the end-effector are checked with the given coordinates for the accuracy of the task. |
5. Conclusion
The article proposes a methodology for testing the developed program of the control system of multi-link robotic arms. This method is based on checking accuracy and repeatability. The modeling is carried out in Matlab. The testing using a robot simulator and a robot. The modeling and testing is based on a computer model for constructing control of multi-link robot arms. The approach proposed in the article can be extended to the task of predicting the self-collision of manipulators [8] and predicting their collision with external objects. Also, the proposed approach can be applied to testing the control system of robot arms based on the “Programming by Demonstration” approach [9].

Acknowledgments
Studies were supported by the Ministry of Science and Higher Education of the Russian Federation (agreement № 14.578.21.0264, project RFMEFI57818X0264).

References
[1] Dhillon B S 1991 Robot Testing and Information Related to Robots. Robot Reliability and Safety Springer, New York.
[2] Wodzinski M, 1987 Robotics Today June 17–20.
[3] Ranky P G, Wodzinski M, 1987 Report No. RSD-TR-1-87, University of Michigan February.
[4] Warnecke H J, 1977 Proceedings of the 7th International Symposium on Industrial Robotics October 10–15.
[5] Ranky P G, Ho C Y, Robot Modelling: Control and Applications with Software, Springer-Verlag, Berlin, 1985.
[6] Kramar V A, Alchakov V V 2018 Proceedings of the 29th DAAAM International Symposium 0095-0101.
[7] Corke P I, 2017 Robotics, vision and control fundamental algorithms in Matlab Springer International Publishing AG.
[8] Kabanov A A, Tokarev D A, 2020 IOP Conference Series: Materials Science and Engineering 709, ID 044021.
[9] Alchakov V, Kramar V, Larionenko A, 2020 IOP Conf. Series: Materials Science and Engineering 709 ID 022092.