Development and Implementation of Grasp Algorithm for Humanoid Robot AR-601M

Kamil Khusnutdinov[1][a], Artur Sagitov[1][b], Ayrat Yakupov[1][c], Roman Meshcheryakov[2][d], Kuo-Hsien Hsia[3][e], Edgar A. Martinez-Garcia[4][f] and Evgeni Magid[1][g]

[1] *Department of Intelligent Robotics, Higher Institute for Information Technology and Intelligent Systems, Kazan Federal University, 35 Kremlyovskaya street, Kazan, Russian Federation*

[2] *V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, 65 Profsoyuznaya street, Moscow, 117997, Russian Federation*

[3] *Department of Electrical Engineering, Far East University, Zhonghua Road 49, Xinshi District, Tainan City, Taiwan*

[4] *Universidad Autonoma de Ciudad Juarez, Cd. Juarez Chihuahua, 32310, Mexico*

Keywords: Algorithm, Grasp Planning, Hand Pose Detection, Humanoid Robot.

Abstract: In robot manipulator control, grasping different types of objects is an important task, but despite being a subject of many studies, there is still no universal approach. A humanoid robot arm end-effector has a significantly more complicated structure than the one of an industrial manipulator. It complicates a process of object grasping, but could possibly make it more robust and stable. A success of grasping strongly depends on a method of determining an object shape and a manipulator grasping procedure. Combining these factors turns object grasping by a humanoid into an interesting and versatile control problem. This paper presents a grasping algorithm for AR-601M humanoid arm with mimic joints in the hand that utilizes the simplicity of an antipodal grasp and satisfies force closure condition. The algorithm was tested in Gazebo simulation with sample objects that were modeled after selected household items.

1 INTRODUCTION

In the field of manipulator control a process of grasping an object is a serious research problem. There is a growing interest in finding solutions of this problem that could be implemented for any humanoid robot. The area of humanoid robots application is extensive as they can work in variety of environments, including factories or social events. A sheer variety of objects that need to be manipulated by humanoids in different environments is daunting, considering differences in objects characteristics, such as shape, size and weight.

In order to successfully grasp an object, a robot hand must adapt to particular characteristics of the object. Also a robot need to consider environmental conditions when performing grasping actions. Clutter in a grasping area (Zhu et al., 2014) hinders object detection, correct pose estimation and evaluating characteristics of the object's surface. In addition, it is required to consider a possibility that surface characteristics of an object may change under some conditions. For example, it may become wet or be deformed due to compression of the object by robot fingers. In such cases, interaction between the robot hand and the object surface may change sporadically. These limitations should be considered during planning of a grasping action. A solution of objects grasping by a humanoid should include the following steps: obtaining information about a target object, evaluating grasping position(s), and planning movements (possibly, solving inverse kinematics problems for fingers and hand).

Usually, a grasping scene is represented as a 3D point cloud, which is further used to extract various data about a target object (Lippiello et al., 2013). Evaluating position and orientation of the target ob-

[a] https://orcid.org/0000-0001-5699-1294
[b] https://orcid.org/0000-0001-8399-460X
[c] https://orcid.org/0000-0001-9977-2833
[d] https://orcid.org/0000-0002-1129-8434
[e] https://orcid.org/0000-0003-1320-673X
[f] https://orcid.org/0000-0001-9163-8285
[g] https://orcid.org/0000-0001-7316-5664

Khusnutdinov, K., Sagitov, A., Yakupov, A., Meshcheryakov, R., Hsia, K., Martinez-Garcia, E. and Magid, E.
Development and Implementation of Grasp Algorithm for Humanoid Robot AR-601M.
DOI: 10.5220/0007921103790386
In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019)*, pages 379-386
ISBN: 978-989-758-380-3

ject can help determining a type of grasp routine to be used (Huang et al., 2013). In addition, performing shape estimation enables to use predefined representation of objects, e.g., in a form of geometric primitives (Herzog et al., 2014), and then to use precalculated grasp point(s). Using representation of the target object as a 3D point cloud, simple antipodal grasps could be generated (Pas et al., 2017). Moreover, if a hand is not able to reach a required grasping position, the robot could attempt performing a small pushing/pulling movement in order to relocate an object inside the robot hand's workspace.

Various criteria could be employed for estimating a grasp quality (Roa and Suárez, 2015), (Feix et al., 2013). For example, it could be a degree of fixation rigidity of an object in a hand (Chalon et al., 2013). A grasp should be stable and robust, i.e., an object should not move freely in a grasping hand. Additionally, grasping and manipulation should always consider a possibility of occlusion (Romero et al., 2013).

This paper presents a grasp algorithm for humanoid robot AR-601M (Magid and Sagitov, 2017), which fingers are constructed with mimic joints. The algorithm provides an antipodal grasp and satisfies force closure condition using mimic joints. It was tested in simulation within Gazebo environment using a 3D model of the right arm of humanoid robot AR-601M. The tests utilized synthetic objects that were constructed using their physical prototypes in real world, which will be further used for the algorithm experimental validation with the real robot.

The rest of the paper is organized as follows. Section 2 presents a literature review. Section 3 describes kinematics of AR-601M humanoid robot arm and its right arm workspace. Section 4 describes the grasp algorithm implementation. Section 5 presents the results of Gazebo simulation using 3D models of AR-601M right arm and the synthetic objects. Finally, we conclude and discuss future work in Section 6.

2 LITERATURE REVIEW

This section briefly familiarizes a reader with grasp planning strategies and approaches, techniques for their implementation and methods of grasp evaluation.

Selecting an optimal grasp from all possible alternatives is a challenging problem and there are a number of approaches. Empirical approaches, for example, are searching for the best grasp in available experimental data utilizing criteria based on target object features. Authors of data-driven based grasp synthesis review revealed interrelation with analytical methods (Bohg et al., 2014) and identified existing grasping open problems. The main difficulty lies in the design of the appropriate structure representing known object grasps in terms of robot perception that will facilitate further search and synthesis. The need for sufficiently big prior data to achieve a high success rate is one of the biggest disadvantages, however, simulation can be used to generate necessary data.

To solve the problem of generating a stable and robust grasp, authors in (Lin and Sun, 2015) presented an approach to a grasp planning that can reconstruct a simplified human grasp strategy (represented by grasp type and thumb positioning) observing human's actions in similar manipulation. A learned strategy is a represention of a recipe to do manipulation with objects of a particular geometry. The integration of such strategies into grasp planning procedure acts as a constraint on a search space, thus allowing planning to be computed much faster, still providing sufficient space not restricting arm agility. A resulting approach integrating a set of learned strategies was compared with the GraspIt! grasp planner, which doesn't utilize similar constraints. A comparison between approaches showed that the proposed approach generates grasps much faster. Generated grasps were similar in configurations to human operations. Paper didn't include tests on novel objects, therefore it is not possible to determine if the approach can be extended to a new class of objects.

Hand trajectories, captured during object grasping tasks by human subjects, was used to define and evaluate a set of indicators that were further used to determine and transfer optimal grasp to robot hand(Cordella et al., 2014). Based on the Nelder–Mead simplex method indicators estimates the optimal grasp configuration for a robotic hand, considering the limitations that arise when determining the grasp configuration. Using cross-cylinder as a target object grasp task was executed by the six human subjects. The advantages of the proposed algorithm are that it has a reduced computational cost. With its help it possible to identify and extract quantitative indicators to describe the optimal grasp poses and their reproduction by humanoid robot's hand. It can predict the final hand position after the movement and the optimal configuration of the fingers for grasp execution as soon as it is provided with information about the size of an object and its location. Its disadvantage is that it strongly depends on the similarity of the robotic hand with the human hand, and is that it does not consider possible slippage of the hand during the grasp an object.

The paper (Bullock et al., 2013) presents a classification scheme for humanoid arm manipulations.

Taxonomy is defined on various manipulation behaviors according to the nature of the contact with external objects and the movement of an object. It allows defining simple criteria that can be applied together in order to easily classify a wide range of manipulation behavior for any system in which the hand can be defined. It is argued that the dexterous movements of the hand can offer an expanded workspace manipulation and improved accuracy with reduced energy consumption but at the cost of added complexity. The advantage of the proposed classification scheme is that it creates a descriptive structure that can be used to effectively describe hand movements during manipulation in various contexts and can be combined with existing object-oriented or other taxonomies to provide a complete description of a particular manipulation task. Its disadvantage is that during the manipulations it implies the obligatory movement of the hand and does not consider situations in which the manipulations can be carried out by the movement of the fingers only, with a fixed hand.

To solve the problem of analyzing the movement of arms, paper (Wörgötter et al., 2013) presented an ontology tree for manipulation tasks based on sequences of graphs where each graph represents the relationship between various manipulated objects expressed in adjacency to each other. The ontology tree can also be used as a powerful abstraction used in robotic applications to represent complex manipulation as a set of simple actions (called a chain of semantic events) The advantage of presented ontology is that it allowed determining about 30 types of fundamental manipulations, obtained as a result of an attempt to structure manipulations in space and time. Its disadvantage lies in the fact that to determine the similarity of types of manipulations, an average threshold value of similarity of species equal to 65% was used, which generally demonstrates not high accuracy in determining the difference between species.

The article (Dafle et al., 2014) proposed 12 possible types of grasps, with the help of which it is possible to carry out manipulations under the influence of external forces. An object is manipulated through precisely controlled fingertip contacts, considering non-hand resources. The advantage of the proposed solution is that it allows carrying out manipulations in conditions close to real conditions, in comparison with situations in which the manipulation is carried out with static hands. And in that it allows looking at the process of making manipulations from a different point of view, going beyond the traditionally considered clever manipulations. The disadvantage is that among the presented types of grasp there are no grasps identical to human grasps.

During the generation of grasps, there can encounter with a problem of determining a grasp strategy that is capable of ensuring the compatibility of an object definition tasks and the implementation of its grasp, as well as capable of ensuring adaptability to new objects. To solve this problem, an investigation (Sahbani et al., 2012) of analytical and empirical approaches to the construction of a grasp strategy presents a review of algorithms for synthesizing the grasp of three-dimensional objects. The advantage of the article is that its authors managed to find a possible problems solution that consists in introducing into the work of the robot the ability to autonomously identify the signs of a new object, with the help of which it can understand what object is in front of it. Its disadvantages include the absence of various simulations of grasp using information about the cases in which each of the approaches is most applicable.

An approach (Pham et al., 2015) for evaluating contact forces based only on visual input data provided by a single RGB-D camera aims to solve the problem of estimating forces applied by hand to an object. The input information is extracted using visual tracking of the hand and an object to assess their position during the manipulation. After that, the kinematics of hand movement is calculated using a new class of numerical differentiation operators. Further the estimated kinematics is fed into the program that returns the desired result: the minimum distribution of force along with an explanation of the observed movement. The advantage of the proposed approach is the ability of solving the problem of determining the contact points of a hand and an object when strong occlusions occurred, using an approach based on the assumption that the closed fingers remain in their last observed position until they are visible again. Its disadvantage is that it cannot solve the problem of determining contact points when performing clever manipulation, with moving a finger or sliding, due to the fact that it uses the above presented assumption that is not fair in the case of dexterous manipulations.

In a multitude of manipulation scenarios, a robot may come into collision with objects of unknown shape: as they rotate, they will remain symmetrical. For such objects, there is no known three-dimensional model. The problem arises of assessing the 3D posture and the shape of such objects, which prevents one from understanding what this object is. To solve this problem, the paper (Phillips et al., 2015) proposes an algorithm for the simultaneous evaluation of the posture and shape of an object without using cross-sections. The solution uses the properties of the projective geometry of the surface of revolution. It restores the three-dimensional pose and shape of an object with an unknown surface of rotation from two points of view: suitable types of known, relative orientation. The advantages of the proposed algorithm are that it can work even when only one of the two visible contours of the surface of revolution and that it is suitable for determining the posture and shape of transparent objects, providing clear contours of such objects. Information about whether the algorithm can estimate the position of an object of a similar object and restore its shape from noisy images is not provided.

During the manipulation, one potential problem is the arrangement of feedback acquisition from the hand. To solve this problem, the authors of the paper (Cai et al., 2016) presented the hypothesis that for accurate recognition of manipulations, it is necessary to model the types of hands and attributes of objects being manipulated. The paper presents a unified model for evaluating the manipulation of the hand and an object, in which the observation of the manipulation is performed from a wearable camera. From the areas of the hand detected on one image, the type of grasp is recognized, and its attributes are determined from the detected parts of an object. The nature of the manipulation is determined by the relationship between the type of grasp and the attributes of an object, representing a set of beliefs about them embedded in the model. The paper provides a model estimate for the correlation between the type of grasp and shape of an object. The advantage of the model is that it exceeds the traditional model that does not consider the interrelation of such semantic constraints as the type of grasp and the attributes of an object. Its disadvantage is that it has an average recognition accuracy of the type of grasp at 61.2%.

3 AR-601M ARM KINEMATICS

Each arm of AR-601M humanoid robot has 20 degrees of freedom (DOF), where 7 DOFs correspond to the arm and 13 DOFs correspond to the five fingers of an arm. The fingers are designed with mimic joints in all phalanges but proximal. The 3D robot model is constructed in a Gazebo simulator environment (Shimchik et al., 2016). Figure 1 demonstrates a 3D model of AR-601M humanoid right arm, which is an exact replica of the real robot right arm. The robot model was integrated into the Robot Operating system (ROS) and MoveIt! motion planner framework. RRTConnect algorithm was chosen for trajectories planning in control of AR-601M arm movements (Lavrenov and Zakiev, 2017).

The arm movement planning requires solving inverse kinematics (IK) problem first. Since ROS contains a set of IK solvers, the solution of IK problem was reduced to a suitable plugin selection that suited our robot's arm constraints and desired characteristics. We selected *kdl_kinematics_plugin*, which is an effective tool for solving IK for manipulators with 6 or more DOFs. Figure 2 shows a pre-grasp position that was calculated with *kdl_kinematics_plugin*.

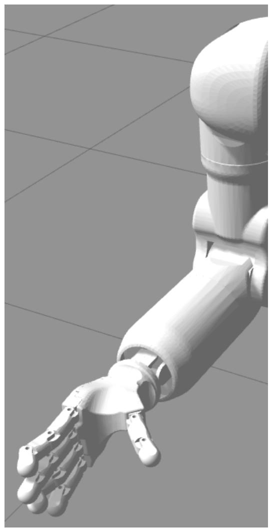


Figure 1: A 3D model of AR-601M humanoid right arm.

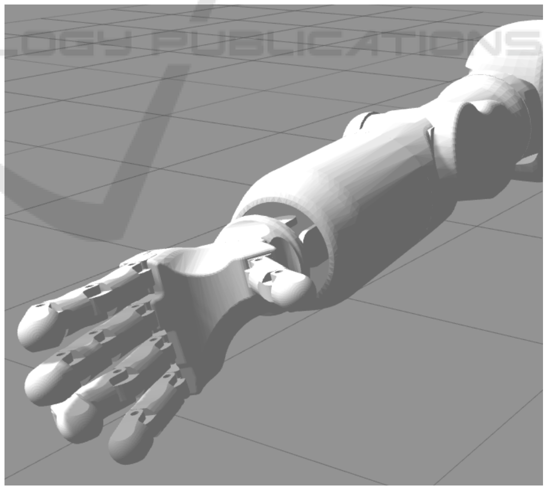


Figure 2: Pre-grasp position of AR-601M arm.

To apply the 3D model of AR-601M humanoid right arm it is necessary to calculate its workspace in order to prepare a scene and calculate the arm movements. The workspace was calculated numerically through applying series of forward kinematics cases until the representative density of workspace reachable (by the end effector) points was achieved.

We used Matlab environment and Robotics Toolbox (Corke, 2017) to calculate reachable workspace of AR-601M right arm (Fig. 3).

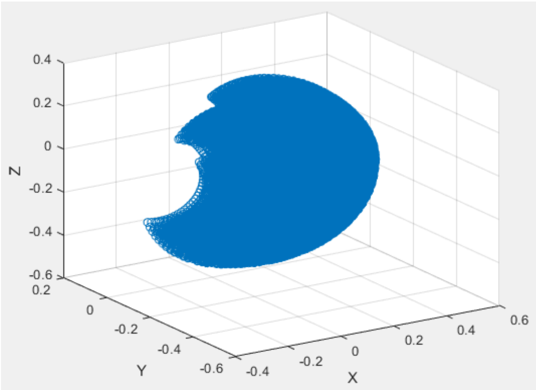


Figure 3: Reachable workspace of AR-601M right arm that was calculated in Matlab environment.

4 GRASPING ALGORITHM

Because of designing a humanoid hand as a set of fingers, some grasp algorithms for humanoids include calculation of contact points (Yu et al., 2017), (V Le et al., 2010). For our algorithm we assume that it is possible to execute a reliable grasp using simple approach without calculating contact points, since it is enough to apply basic forces to an object surface within the plane (similarly to antipodal grasp execution). At this stage, an object may not be rigidly fixed in the hand yet, but it will not fall out of the grasp. Next, the robot completes the grasp applying additional forces and performing flexion of the remaining fingers. The advantage of the algorithm is that it avoids contact points calculation. Its disadvantage is that depending on a generated handle location, the flexion of the ring and the pinkie fingers could be execute outside of the object. Therefore, it is not always possible to satisfy the force closure condition.

4.1 Obtaining Object Data from a Point Cloud

Before grasping an object it is necessary to find out information that will allow the grasping. To obtain such information various tools could be used, but most of these tools employ a 3D point cloud as a source for further data extraction. The idea behind is that objects and a scene of manipulation are represented as a combination of a large number of 3D points that are located very close to each other. A point cloud could be obtained with RGBD cameras, image depth sensors or motion sensors, which are used as image depth sensors. They allow receiving both a color image and its 3D representation in a form of a 3D point cloud. Figure 4 shows a point cloud for 3D models of objects (a bottle and a ladle) that were obtained with Microsoft Kinect sensor within Gazebo simulator.

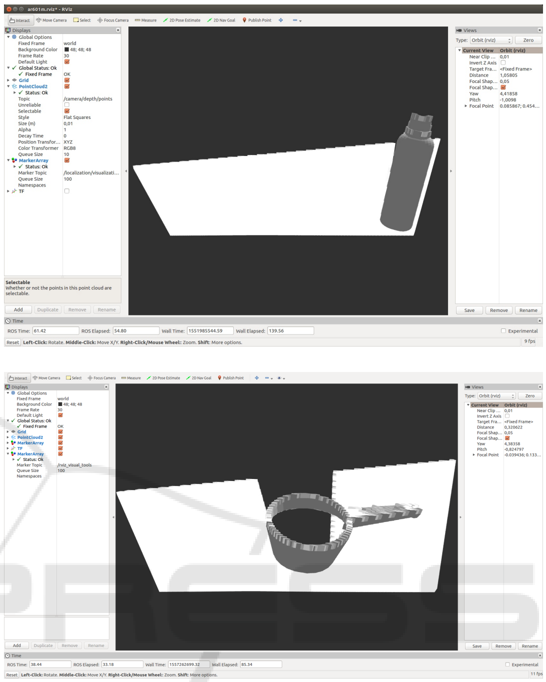


Figure 4: 3D models of bottle (top) and ladle (bottom) represented as 3D point clouds.

4.2 Objects Description and Grasp Geometry

Various tools that allow extracting data (which is necessary in order to grasp an object) from a 3D point cloud, could provide data about contact points and their coordinates in 3D space, an area from which an object should be extracted, and objects geometry. Next, the extracted data is used for grasp executing. Some tools can generate grasps, but typically these are simple antipodal grasps, which are carried out by a two-finger gripper with the fingers moving in the same plane and driven by a prismatic joint. ROS package *handle_detector* (Pas and Platt, 2014) is a tool that extracts data about an object from a scene by analyzing objects surface (represented as a 3D point cloud), identifies and visualizes an area of the object (referred as a handle) that should be grasped. Each handle is represented with a set of cylinders with their own parameters, providing handle-related data that includes handle position, orientation, radius and extent. Figure 5 shows handles for 3D models of a bottle and a ladle that were generated using *handle_detector* tool.

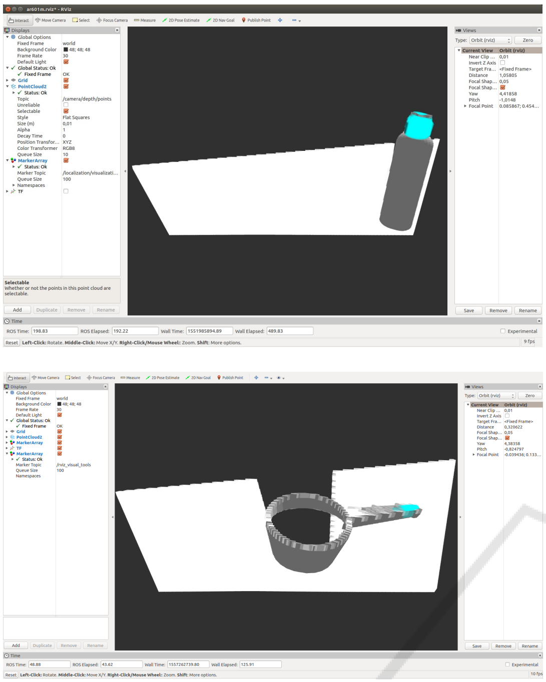


Figure 5: Handles for 3D models of a bottle (top) and a ladle (bottom) are shown with cyan color.

4.3 Calculating a Target Pose of an Object Grasping Hand

Before grasping an object it is necessary to find a target point for a hand that it needs to reach in order to perform the antipodal grasp. When the hand reaches its target position, an object should lie between the thumb and the other fingers. The index finger was selected as a pair for the thumb in the antipodal grasp. With *handle_detector* tool this corresponds to a situation where the hand reaches its target point, and thus handle center coordinates (point A) should align with point B. To implement this idea, the coordinates of thumb tip (point C) and index tip (point D) in their default positions (an open antipodal gripper) are determined. Next, coordinates of point B are determined and subtracted from coordinates of point A in order to calculate a displacement vector. To determine coordinates of the hand target point, the displacement vector coordinates are added to the coordinates of the current hand position. Figure 6 visualizes these antipodal grasp geometry calculation.

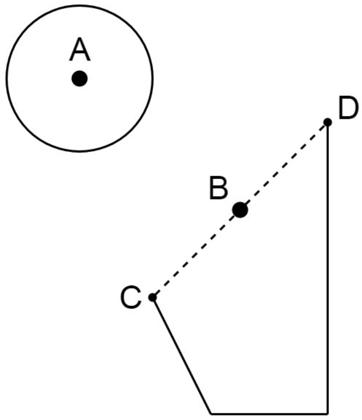


Figure 6: Components for calculation a target point for a hand. A is a center of a handle, B is a point that is equidistant from tips of a thumb and an index fingers, C is a tip of the thumb, D is a tip of the index finger.

4.4 Determining Positions of Fingers to Grasp an Object

Each finger tip position is determined with calculations of finger joints' rotation angles. Since the joints are mimic, this reduces to calculating rotation angles for active joint of a finger, which is a first joint. Using *handle_detector* tool we obtain diameter d of a handle's cylinder. Next, thumb and index fingers flexing is simulated and distance S between their tips is tracked and compared to d, resulting into optimal mimic joint rotation angles selection that minimizes the difference between d and S. To calculate mimic joint rotation angles of other fingers, it is necessary to consistently increase the angles of their active joints and select the ones that help satisfying the force closure condition. Figure 7 visualizes these calculations.

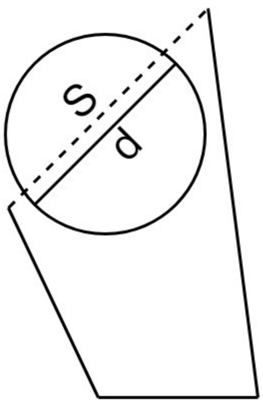


Figure 7: Calculating rotation angles: d is a handle diameter, S is a distance between thumb and index tips.

5 SIMULATION OF GRASPING AN OBJECT

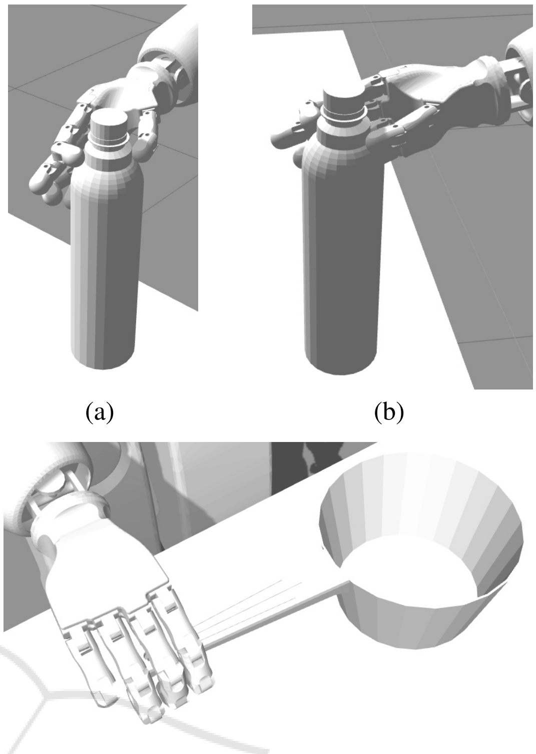
The grasp algorithm for the AR-601M humanoid using *handle_detector* tool consists of the following sequence of actions:

1. Estimate coordinates of a handle's center;
2. Use coordinates of fingertips that are in an open state to calculate a point that is equidistant from the fingertips;
3. Subtract coordinates of the point founded in Step 2 from the handle's center coordinates, find the hand offset vector coordinates;
4. Add the coordinates of Step 3 to current hand position coordinates, find coordinates of a new hand target position that is necessary to grasp an object;
5. Estimate trajectory of grasp fingers' flexion to calculate distances between fingertips and the handle's circumference. Active joints' angles that are minimizing the difference between the distance between the fingertips and the handle circumference are the required angles of flexion required to grasp the object.

The algorithm was verified in virtual experiments that were performed in Gazebo simulation. Five different models of real world objects were tested: a bottle, a rectangular box of vitamins, a juice box, a ladle and a plastic cup without a handle. Figure 8 shows virtual experiments of grasping 3D models of a bottle and a ladle by AR-601M humanoid robot using the proposed grasping algorithm.

6 CONCLUSION AND FUTURE WORK

We presented the development and implementation of a grasping algorithm for AR-601M humanoid robot that utilizes a simplicity of an antipodal grasp and satisfies force closure condition using mimic joints. The algorithm was tested in Gazebo simulation environment with five different synthetic objects that were constructed using their physical prototypes in real world. Our approach's advantages and disadvantages were discussed. As a part of our future work we plan to validate the algorithm in real world environment with AR-601M humanoid for execution of pick and place operations. RGBD sensor will be used to provide the 3D point cloud data for the algorithm. One of the particular tasks that will be implemented using the proposed algorithm is door handle grasping and door opening. This is a necessary task of AR-601M humanoid in victim search mission within household environment that is a part of our large-scale project in robotized urban search and rescue.



(a) (b) (c)

Figure 8: Grasping objects with the proposed algorithm (a), (b) a bottle; (c) a ladle.

ACKNOWLEDGEMENTS

The reported study was funded by the Russian Foundation for Basic Research (RFBR) according to the research project No. 19-58-70002. Part of the work was performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

REFERENCES

Bohg, J., Morales, A., Asfour, T., and Kragic, D. (2014). Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, 30(2):289–309.

Bullock, I. M., Ma, R. R., and Dollar, A. M. (2013). A hand-centric classification of human and robot dexterous manipulation. *IEEE Transactions on Haptics*, 6(2):129–144.

Cai, M., Kitani, K., and Sato, Y. (2016). Understanding hand-object manipulation with grasp types and object attributes. In *Robotics conference: Science and Systems 2016*.

Chalon, M., Reinecke, J., and Pfanne, M. (2013). Online in-hand object localization. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2977–2984.

Cordella, F., Zollo, L., Salerno, A., Accoto, D., Guglielmelli, E., and Siciliano, B. (2014). Human hand motion analysis and synthesis of optimal power grasps for a robotic hand. *International Journal of Advanced Robotic Systems*, 11(3):37.

Corke, P. I. (2017). *Robotics, Vision & Control: Fundamental Algorithms in MATLAB*. Springer, second edition. ISBN 978-3-319-54412-0.

Dafle, N. C., Rodriguez, A., Paolini, R., Tang, B., Srinivasa, S. S., Erdmann, M., Mason, M. T., Lundberg, I., Staab, H., and Fuhlbrigge, T. (2014). Extrinsic dexterity: In-hand manipulation with external forces. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1578–1585.

Feix, T., Romero, J., Ek, C. H., Schmiedmayer, H., and Kragic, D. (2013). A metric for comparing the anthropomorphic motion capability of artificial hands. *IEEE Transactions on Robotics*, 29(1):82–93.

Herzog, A., Pastor, P., Kalakrishnan, M., Righetti, L., Bohg, J., Asfour, T., and Schaal, S. (2014). Learning of grasp selection based on shape-templates. *Auton. Robots*, 36:51–65.

Huang, B., El-Khoury, S., Li, M., Bryson, J. J., and Billard, A. (2013). Learning a real time grasping strategy. In *2013 IEEE International Conference on Robotics and Automation*, pages 593–600.

Lavrenov, R. and Zakiev, A. (2017). Tool for 3d gazebo map construction from arbitrary images and laser scans. In *2017 10th International Conference on Developments in eSystems Engineering (DeSE)*, pages 256–261. IEEE.

Lin, Y. and Sun, Y. (2015). Robot grasp planning based on demonstrated grasp strategies. *The International Journal of Robotics Research*, 34:26–42.

Lippiello, V., Ruggiero, F., Siciliano, B., and Villani, L. (2013). Visual grasp planning for unknown objects using a multifingered robotic hand. *IEEE/ASME Transactions on Mechatronics*, 18(3):1050–1059.

Magid, E. and Sagitov, A. (2017). Towards robot fall detection and management for russian humanoid ar-601. In *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, pages 200–209. Springer.

Pas, A., Gualtieri, M., Saenko, K., and Platt, R. (2017). Grasp pose detection in point clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473.

Pas, A. and Platt, R. (2014). Localizing handle-like grasp affordances in 3d point clouds. In *The 14th International Symposium on Experimental Robotics*, volume 109.

Pham, T.-H., Kheddar, A., Qammaz, A., and Argyros, A. A. (2015). Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2810–2819.

Phillips, C. J., Lecce, M., Davis, C., and Daniilidis, K. (2015). Grasping surfaces of revolution: Simultaneous pose and shape recovery from two views. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1352–1359.

Roa, M. A. and Suárez, R. (2015). Grasp quality measures: review and performance. *Autonomous Robots*, 38(1):65–88.

Romero, J., Kjellström, H., Ek, C. H., and Kragic, D. (2013). Non-parametric hand pose estimation with object context. *Image and Vision Computing*, 31:555–564.

Sahbani, A., El-Khoury, S., and Bidaud, P. (2012). An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60:326–336.

Shimchik, I., Sagitov, A., Afanasyev, I., Matsuno, F., and Magid, E. (2016). Golf cart prototype development and navigation simulation using ros and gazebo. In *MATEC Web of Conferences*, volume 75, page 09005. EDP Sciences.

V Le, Q., Kamm, D., F Kara, A., and Y Ng, A. (2010). Learning to grasp objects with multiple contact points. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5062 – 5069.

Wörgötter, F., Aksoy, E. E., Krüger, N., Piater, J., Ude, A., and Tamosiunaite, M. (2013). A simple ontology of manipulation actions based on hand-object relations. *IEEE Transactions on Autonomous Mental Development*, 5(2):117–134.

Yu, D., Yu, Z., Zhou, Q., Chen, X., Zhong, J., Zhang, W., Qin, M., Zhu, M., Ming, A., and Huang, Q. (2017). Grasp optimization with constraint of contact points number for a humanoid hand. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2205–2211.

Zhu, M., G. Derpanis, K., Yang, Y., Brahmbhatt, S., Zhang, M., Phillips, C., Lecce, M., and Daniilidis, K. (2014). Single image 3d object detection and pose estimation for grasping. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3936–3943.