Simulation of Grasp Localization Inferences for a Dual-Arm Collaborative Robot

Josin Hippolitus A¹*, R.Senthilnathan¹, Omar Malla¹
¹Department of Mechatronics Engineering, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, 603203, Kanchipuram, Chennai, TN, India.

E-mail : josinhippolitus@gmail.com

Abstract. Grasp Localization is an important problem in robotic manipulators which fundamentally aims at estimating a pose for the end effector which ensures that the object intended to be grasped does not slip off. This estimation is achieved through exteroceptive sensors used in robots such as vision systems which through features and learning-based approaches. One of the fundamental means to validate such an approach is by simulating the grasp routine for any given geometry and its probable grasp poses. This paper presents the details of a simulation work involving a dual-arm collaborative robot with each arm consisting of seven degrees of freedom. The dual-arm robot performs a grasp routine for a geometry that is imported from CAD software. To perform the simulation the various attributes of the robot are to be observed since the simulated robot is a mimic of a commercially available one.

Keywords: deep learning, grasp localization, collaborative robot

1. Introduction
Grasping is an important activity in manipulator robotics. It is the first step of a robot interacting with an object. The end-effector or the tool center point (TCP) frame, typically in a gripper scenario, must be guided to a suitable pose on the object to be grasped. When the geometry of the object is unknown, computer vision systems may be used to identify the right pose. The best grasp is achieved when the gripper successfully picks the object without any slip. To find the perfect grasp, grasp synthesis is performed, where deep learning is used [5]. For a given environment there can be any object with known geometry or unknown geometry. In model based approaches, the success rate of the grasp is very high, as the geometry of the object is known. Ever since a gripper is in use, analytical methods and expert knowledge in a certain application area has driven so far to execute a successful grasp. Nevertheless to say that they are most time consuming and least generalized. But when the robot is subjected to grasp a novel object, vision and touch based approaches are the more preferred methods to make the grasp successful. The cameras and tactile sensors help in providing the information about those novel objects subjected for grasp. Convolutional Neural Networks are employed to extract the features and identify the areas of interest for grasping using data driven methods. Here deep learning techniques are used for grasp localization.

2. Challenges, approaches for GL
In robotic grasp planning the major challenges observed are the tedious work on labelling the grasps by expert engineers and their biased nature on grasp localization. The biased nature on grasp localization is due to the nature of human beings’ understanding of semantics and their experiences in the past. By automating the process of labelling, by reinforcement learning both challenges could be overcome.

A robotic gripper has to be aligned properly in an image plane according to the pose of the object. First the image plane coordinates has to be aligned with the robot world coordinates as a part of grasp
planning. The features of the novel object are marked during this process. Based on the detected features, the control system guided by a control algorithm will control the joints of the robotic manipulator to reach the object, also the end effector to orient itself to perform the grasp.

3. Deep learning for GL
The input data is an RGB-D image, which on using a convolutional neural network provides features of the objects present in the field of view. On further processing with deep neural networks, they yield multiple grasp configurations. The best grasp among them is found and is ranked as positive and negative grasps. Representation of RGB and depth image with rectangles is done by Ian Lenz et al., in 2014 [4]. Redmon et al. 2015 [7] introduced one-shot detection method which showcased a detection accuracy of 88% and detection speed of 76 milliseconds for an image. The decision making representational heuristics were obtained from the existing data by performing mathematical operations simultaneously and parallel. Normally, two-fingered grippers are used for grasping. But for challenging objects like a sphere, multi-fingered grippers are preferred [6].

4. DL approaches
The most common networks used for convolutional neural networks are AlexNet and ResNet. Among them ResNet has been identified as a better performing network than AlexNet. The ResNet has skip connections, which could skip unnecessary weights and biases to provide efficient results. Recent researches focus on PointNet, which takes the point cloud data directly and process the same. By doing so, the speed has been increased to a larger extent. Kumra et al. [8] implemented a much deeper network model and obtained detection accuracy upto 89.21%. Vierreck et al. [2] Used (x,y,θ) and depth image as input and distance as output, which could tackle unexpected disturbances. Schwarz et al. [3] used two deep-learning approaches for object detection and semantic segmentation and one item model registration method; hence the system localized the requested item. Representation of RGB and Depth image with rectangles, Ian Lenz et al. [4] presented a method to apply structured regularization on the weights which is based on multimodal group regularization. Hence, multimodal inputs will be handled efficiently. He also used two deep networks to present a two-step cascaded system. In this method significant detections from the first were re-evaluated by the second. The first network is faster since it has fewer features and can effectively eliminate inefficient candidate grasps. The second is slower since it has much more features. But it finds its significance as it will only evaluate only significant detections.

LaserNet [9] is also a focus of research in recent times. LaserNet is the only probabilistic model, which predicts the probabilities of the dimensions of the bounding box. This has increased the detection performance as a whole. LaserNet works on the point cloud data obtained from LiDARs, to provide bounding boxes on the same. The predicted variance is used to determine an appropriate IoU (Intersection over Union) threshold for a pair of boxes. The likelihood of the box is used as its score [9].

Figure 1. Illustration of the adaptive NMS technique [9]
During reinforcement learning of grasp localization, the success of the grasp is verified by few parameters: how successfully the gripper approaches and reaches the object in the desired pose, and how successfully it picks up the object without any slippage. When the gripper holds the object in the wrong location, then the possibility of the object falling is very high. Only when the object is being held in the right location, the grasp becomes successful. In reinforcement learning, the grasps are ranked, and the best ranked is selected based on a state of art method available and executed. The two-stage detection process of grasp rectangle and predicting the perfect grasp [4].

Johns et.al. [1] used a method that first predicts a score for every possible grasp pose. For the same, he created a grasp function. Within the scope of the paper no deep learning network is explicitly presented, though deep learning based grasp location is the current trend. The paper only highlights the simulation methodology after getting inferences from deep learning networks.

5. Importance of Simulation for GL
Reinforcement learning is a rigorous and time-consuming process by itself. Hence, we have to wait for a lot of time to find the result. But when we adopt simulation for doing the same, we could achieve quicker results. Mahler et.al [10] suggested using grasp simulation as one of the physics-based analyses to populate data when needed.

6. Simulation practices in grasp localization
There is also another handful number of researchers who used simulation for populating data. John et.al used 3D meshes for depth image and physics simulation [1]. Bohg et.al formed empirical methods using simulations on real robotic systems [2]. Amazon robotics also generated training data using simulations [11]. On the other hand, simulation practices for grasping are also a common practice nowadays [12]. [13] grasp is a simulation technique to compute grasp contacts using a robotic hand. They used Columbia Grasp Database. A soft finger contact model used GraspIt for simulating their robot [14]. K. Bousmalis et al used GraspGAN methodology [15] to obtain grasping performance equivalent to real-time grasping performance. The key to note that they obtained this without any real-world labels. OpenGRASP [16] is another toolkit used for robot grasping simulation which has extensibility and interoperability. [17] PyBullet is a more common simulator used for the simulation of grasping. In few grasp simulation projects [18] Q function estimation method is used for benchmarking the tasks.

In this project, grasp localization is simulated by the approach of the robot by generating simulated grasp pose for the end-effector. The simulated grasp pose is considered to be the ideal pose which would ensure that the object does not slip. The grasping simulation is performed by simultaneously applying the transformation which the end-effector is undergoing from time-to-time, to the object. This creates a visualization of the object being held by the robot and moved around.

7. 7-DOF Dual-arm Collaborative Manipulator
In this new era of collaborative work handling between machines and human beings, collaborative robots find its significance is not only material and tool handling, but also grasping, pick and place and many more serviceable arenas. Yi Ren et.al [19] presented a novel bio-mimetic object impedance control strategy for dual-arm co-operative 7 Degrees Of Freedom Humanoid manipulators. This manipulator exhibits human-like character, namely explicit compliance behavior during weak environment interaction and vice versa.
Jun He [20] proposed a dual loop impedance control which exhibits adaptive stiffness for a dual-arm cooperative robot. This controller is used for grasping a given object in the environment.

In 2015 ABB has introduced YuMi, a 7 axis agile collaborative robot [21] which enabled people to work around and along with the robot.
8. Specification of the physical manipulator
The manipulator of ABB YuMi consists of a 7-axis configuration, which has reachability of 0.559 m and a payload capacity of 0.5 Kg. Its position repeatability of 0.02 mm makes it the best choice for grasp localization. Dudek et.al [23] have specified in his work about the dimensions of a manipulator, which is shown in Figure 5. Figure 6 shows the simulated dual-arm YuMi robot.

Figure 5. Specifications of the ABB YuMi dual-arm collaborative robot [23]

9. D-H parameters
D-H parameters are used to represent the architecture of a robot. Usually, they are provided with the robot itself by the manufacturer.

The D-H parameters of ABB YuMi is given here:

| Articulation | $\alpha_i$ (rad) | $d_i$ (cm) | $r_i$ (cm) | $\theta_i$ (rad) |
|--------------|-----------------|------------|------------|-----------------|
| 1            | 0               | 0          | 16.6       | $\pi$           |
| 2            | $\pi/2$         | 3          | 0          | $\pi$           |
| 3            | $\pi/2$         | 3          | 25.15      | $\pi$           |
| 4            | -$\pi/2$        | 4.05       | 0          | $\pi/2$         |
| 5            | -$\pi/2$        | 4.05       | 26.5       | $\pi$           |
| 6            | -$\pi/2$        | 2.7        | 0          | $\pi$           |
| 7            | -$\pi/2$        | 2.7        | 3.6        | $\pi$           |

Table 1: D-H parameters of ABB YuMi [22]
10. Results and Discussion

10.1. Forward kinematics, transformation matrices
Kinematic equations are used to calculate the position of the end effector from the joint parameter values. Forward kinematics uses these equations to calculate the position of the end effector to grasp the object in space using these joint parameter values. In other words, a set of joint angles are given and the position of the end effector has to be found.

![Simulated Dual-arm YuMi robot](image1.png)

**Figure 6.** Simulated Dual-arm YuMi robot

10.2. Inverse kinematics, expressions for each joint angle
The joint parameters required to make the end effector move to a particular pose and rotation during the process of grasping can be determined by inverse kinematics. In other words, the position of the end effector is given and the joint parameters have to be found.

10.3 Workspace Analysis

10.3.1 Jacobian
When we have to deal with two different representations of the system, we use jacobian. They provide the relationship between joint velocities and end effector velocities of a robot manipulator. The robot moves from a resting position to the destination to grasp the object in space.

10.3.2 Singularities
As the robot manipulator moves to pick up the object in space, there is a possibility that two or more axes align to be collinearly resulting in undesirable motions and velocities. Wen et.al [24] states that singularities are the configuration the manipulability ellipsoid becomes degenerate.

10.3.3 Workspace plot
The workspace of the dual-arm YuMi robot simulated has been plotted and shown in figure 7.
Figure 7. Workspace of the simulated YuMi robot

10.4 Trajectory Planning
10.4.1 Joint space planning
The joint space has been plotted below in figure 8 and figure 9. Figure 8 shows the joint space of the left arm and figure 9 shows the joint space of the right arm. In the left arm, the second joint and the sixth joint has the maximum displacement, which means that they rotate to the maximum extent. It is also identified that the first and fifth joints are at zero during the starting.

Similarly, in the right arm also the second and sixth joint has maximum displacement. One significant point to note in the right arm is that the fifth joint moves from zero to negative, whereas in the left arm, it moves from zero to positive. The third arm moves from negative to positive in the right arm, whereas it is vice-versa for the left arm.
As the displacement of the second and sixth joint is higher, so the velocity also. This is common in both arms. The velocity of the third joint oscillates in the negative, whereas the velocity of the fifth joint oscillates in the positive. This is vice-versa in the right arm. They are shown in figure 10 and figure 11.

Figure 9. Displacement with time

Figure 10. Velocity with time

Figure 11. Velocity with time
Figure 12 and Figure 13 show the acceleration of the joint angles with time. For both arms, it is being identified that the sixth arm has maximum acceleration, whereas the first arm has minimum acceleration.

10.4.2 Cartesian space planning

When the robot joints are actuated the end-effector moves to reach the target and grasp it. The figures below, figure 14 and figure 15 depict the Cartesian space displacement of the left and right arm. In the left arm, the fourth joint deflects to the maximum in the Cartesian space, whereas in the right arm, the first joint moves up to the edge of Cartesian space.
In order to simulate the grasp localization, the object needs to be imported into the robot simulation environment. The geometries of the objects either are extracted from any sensor such as time-of-flight cameras, LIDAR, etc or from CAD software for the visualization. In this paper the CAD import approach is presented. The CAD data of the object of interest is exported as an ASCII STL file. Most CAD software such as AUTOCAD, ProE, etc supports this feature. A third party tool [25] is utilized for converting the STL file into MATLAB workspace which may be subjected to some rotational transformations for pose presentation with reference to the robot’s location. An arbitrary pose on the object was chosen based on manual observation of the geometry and target pose is created for the robot. A simulation step of the grasp is presented in Figure 16 for a sample object.
Figure 16. The simulation of the robot grasping the object in space is shown here

11. Conclusion
In the field of grasp localization, estimating the pose of the gripper for a successful grasp is a significant challenge. In this paper a grasp routine in seven degrees of freedom dual arm robot is simulated for the given object with known geometries which is imported from CAD software. This will also validate the feature based approaches. The paper discussed the details of the kinematics, workspace analysis, CAD import, planning of trajectories in Cartesian and joint space, the control strategy for executing the trajectory, and ultimately the grasping action. The main contribution of the paper is the simple methodology of simulating the grasp localization and grasping in a standard software platform thereby paving ways for various analysis of the grasp location problem. The results of simulation and the strategies adopted will be useful for technical community working in the field of grasp localization to quickly get started with grasp simulations using a redundant dual-arm manipulator.

References
[1] Johns, E.; Leutenegger, S.; Davison, A.J. Deep learning a grasp function for grasping under gripper pose uncertainty. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4461–4468.
[2] Viereck, U.; Pas, A.; Saenko, K.; Platt, R. Learning a visuomotor controller for real-world robotic grasping using simulated depth images. In Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, California, USA, 13–15 November 2017; Volume 78, pp. 291–300
[3] Schwarz, M.; Milan, A.; Lenz, C.; Muñoz, A.; Periyasamy, A.S.; Schreiber, M.; Schüller, S.; Behnke, S. Nimbro Picking: Versatile Part Handling for Warehouse Automation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017
[4] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps, Learning Methods in Robotic Grasp Detection, Multimodal Technologies and Interact. 2018, 2, 57. arXiv:1301.3592v6.
[5] Shehan Caldera; Alexander Rassau and Douglas Chai. Review of Deep Learning Methods in Robotic Grasp Detection, Multimodal Technologies and Interact. 2018, 2, 57.
[6] S. Carpin, S. Liu, J. Falco, K. Van Wyk, Multi-fingered robotic grasping: A primer, 2016, [online] Available
[7] Redmon, J.; Angelova, A. Real-time grasp detection using convolutional neural networks. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1316–1322.
[8] Kumra, S.; Kanan, C. Robotic grasp detection using deep convolutional neural networks. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 769–776.

[9] Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, Carl K. Wellington; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 12677-12686

[10] Mahler, J.; Liang, J.; Niyaz, S.; Laskey, M.; Doan, R.; Liu, X.; Ojea, J.A.; Goldberg, K. DexNet 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics. CoRR 2017, abs/1703.09312.

[11] Amazon Robotics. Amazon Picking Challenge 2016. https://www.amazonrobotics.com/#/picking challenge. Accessed: 2017-07-06

[12] Bohg, J.; Morales, A.; Asfour, T.; Kragic, D. Data-Driven Grasp Synthesis - A Survey. IEEE Transactions on Robotics 2014, 30, 289–309

[13] A. T. Miller and P. K. Allen, "Graspit! A versatile simulator for robotic grasping," in IEEE Robotics & Automation Magazine, vol. 11, no. 4, pp. 110-122, Dec. 2004, DOI: 10.1109/MRA.2004.1371616.

[14] H. Dang, J. Weisz, and P. K. Allen, "Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics," 2011 IEEE International Conference on Robotics and Automation, Shanghai, 2011, pp. 5917-5922, doi: 10.1109/ICRA.2011.5979679.

[15] K. Bousmalis et al., “Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping,” 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, 2018, pp. 4243-4250, doi: 10.1109/ICRA.2018.8460875.

[16] León B. et al. (2010) OpenGRASP: A Toolkit for Robot Grasping Simulation. In: Ando N., Balakirsky S., Hemker T., Reggiani M., von Stryk O. (eds) Simulation, Modeling, and Programming for Autonomous Robots. SIMPAR 2010. Lecture Notes in Computer Science, vol 6472. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-17319-6_13

[17] E. Coumans and Y. Bai, pybullet a python module for physics simulation games robotics and machine learning, 2016-2017

[18] D. Quillen, E. Jang, O. Nachum, C. Finn, J. Ibarz and S. Levine, "Deep Reinforcement Learning for Vision-Based Robotic Grasping: A Simulated Comparative Evaluation of Off-Policy Methods," 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, 2018, pp. 6284-6291, doi: 10.1109/ICRA.2018.8461039.

[19] Yi Ren, Yechao Liu, Minghe Jin, Hong Liu, Biomimetic object impedance control for dual-arm cooperative 7-DOF manipulators, Robotics and Autonomous Systems, Volume 75, Part B, 2016, Pages 273-287, ISSN 0921-8890.

[20] J. He, M. Luo, and Q. Zhang, "Dual impedance control with variable object stiffness for the dual-arm cooperative manipulators," 2016 Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Tokyo, 2016, pp. 102-108, doi: 10.1109/ACIRS.2016.7556196.

[21] Meryem Taghbalout, Jean François Antoine, Gabriel Abba, Experimental Dynamic Identification of a YuMi Collaborative Robot, IFAC-PapersOnLine, Volume 52, Issue 13, 2019, Pages 1168-1173, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2019.11.354.

[22] Dudek and Jenkin, Computational Principles of Mobile Robotics and CS W4733 – Differential Drive Robots. (https://chess.eecs.berkeley.edu/eecs149/documentation/differentialDrive.pdf)

[23] 24. J. T. -. Wen and L. S. Wilfinger, "Kinematic manipulability of general constrained rigid multibody systems," in IEEE Transactions on Robotics and Automation, vol. 15, no. 3, pp. 558-567, June 1999, doi: 10.1109/70.768187.

[24] Don Riley (2020). CAD2MATDEMO.M (https://www.mathworks.com/matlabcentral/fileexchange/3642-cad2matdemo-m), MATLAB Central File Exchange. Retrieved October 1, 2020.