# Grasping Tools with Task-Oriented Design in Pybullet Simulation

VENKATA NAVEEN PASUMARTHI - V684G963

*Abstract*— **This paper presents a novel approach to robotic grasping that combines the use of the PyBullet physics simulation environment and inverse kinematics algorithms. The goal of the research is to develop a system that can autonomously grasp and manipulate a wide range of objects with varying shapes and sizes.**

## I. INTRODUCTION

In recent years, the field of robotics has advanced significantly, especially in the areas of robot manipulation and grasping.Grasping is a critical function for robots that need to interact with the physical world, such as in manufacturing, logistics, and household robotics.

Pybullet simulates the dynamics of grasping in an exact and realistic manner, allowing us to analyze and refine grasping tactics prior to deploying them on a physical robot.meanwhile,in inverse kinematics techniques allow us to determine the joint angles of a robotic arm that will result in the required end-effector position and orientation, allowing us to construct the best gripping trajectories.

We illustrate the efficacy of our method in a series of studies involving gripping items of various forms, sizes, and orientations.The suggested method has important implications for the development of robotic grasping systems capable of performing complicated manipulation tasks in a variety of areas, including manufacturing and logistics. It also opens up new avenues for study into the interaction of robots with the physical environment, particularly in situations when human engagement is limited or impracticable.

### A. *Background about robotic Grasping*

In the area of robotics known as robotic grasping, researchers work to create algorithms and systems that let machines take up and control items. The objective is to develop robots that are capable of jobs that are challenging or hazardous for humans, such handling delicate materials or working in hazardous locations.

Object detection, motion planning, and control are only a few of the issues that need to be solved in robotic grasping. Finding out about an object's characteristics, such as its shape, size, and material, requires object recognition. Motion planning entails formulating a series of actions that will enable the robot to grip the item, whereas control entails carrying out those actions precisely. Robotic grasping can be accomplished in a variety of ways, such as employing sensors to detect items, machine learning algorithms to identify objects and design grasping motions, and haptic feedback to give the robot a sense of touch. Suction cups, gripper fingers, and magnetic grippers are just a few of the several varieties of grippers that can be utilized. Numerous industries, including manufacturing, logistics, and healthcare, can benefit from robotic grasping. For instance, robots can be employed to help surgeons during operations, sort parcels in a warehouse, or construct things on an assembly line.

### B. *Brief summary of the project*

This project shows how the robot picks up the random object and actual assigned object for at least five seconds.It uses inverse kinematics to operate robot in pybullet environment.It also demonstrates how the object is successfully recognised by the robot based on the tune-grasping readings and grasps the object.PyBullet simulation was used to evaluate and optimize the grasping techniques in a realistic environment.

## II. PROJECT DESCRIPTIONS

### A. *Describe the procedure of the project*

- First we need to get familiar with 3D robot simulation and programming
- Setup the python3 environment on your system.
- Install Pybullet Environment: By using the command pip3 install pybullet –upgrade –user. We install pybullet.
- Run the command in the terminal: We can run the provided grasping demo for testing the Pybullet environment. Run "tune-grasping.py" for testing.
- Use the output results from the tune-grasping.py to design the grasping type.
- Use the output results from the tune-grasping.py to design the grasping type.
- Update the grasping type of the main.py from the readings which we got in tune-grasping.
- Generate URDF file for the second assigned object:
  a. Use Mesh Cleaner to clean the mesh and calculate the center mass and inertia matrix.
  b. https://www.hamzamerzic.info/mesh-cleaner/
  c. Change the value of 'rpy' under ¡inertial¿ tag to the center mass that are calculated by Mesh Cleaner.
  d. Change the value of 'ixx', 'ixy', 'ixz', 'iyy', 'iyz', and 'izz' under ¡inertial¿ tag to the inertia matrix that is calculated by Mesh Cleaner. calculated by Mesh Cleaner.
  e. Download the cleaned mesh file from Mesh Cleaner.
  f. Change the file name of the object under ¡visual¿ and ¡collision¿ tags to the downloaded file.
  g. Change the scales under ¡visual¿ and ¡collision¿ tags.

### B. *Show images of grasping types*

The following images describes the versatility of robotic grasping, showcasing different grasping strategies that can be used depending on the size and shape of the object being grasped.

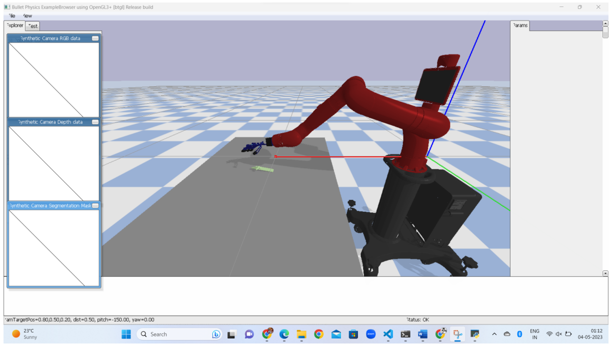

Fig. 1. Robot before grasping the random object

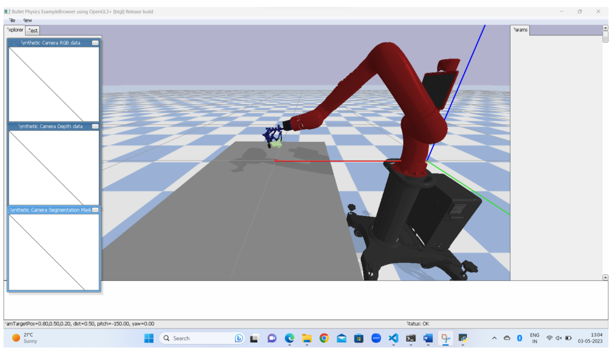

Fig. 2. Robot while grasping the random object

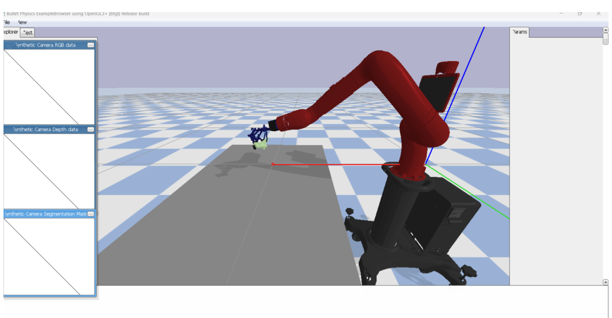

Fig. 3. Robot successfully grasping the random object

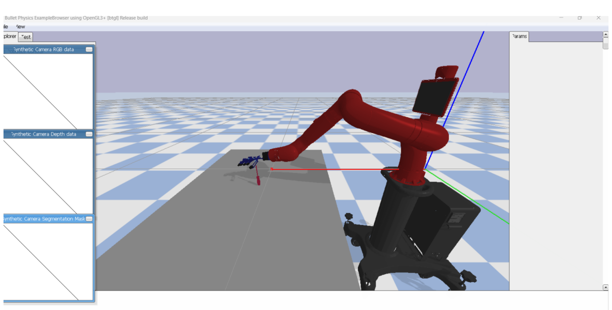

Fig. 4. Robot before grasping the actual object

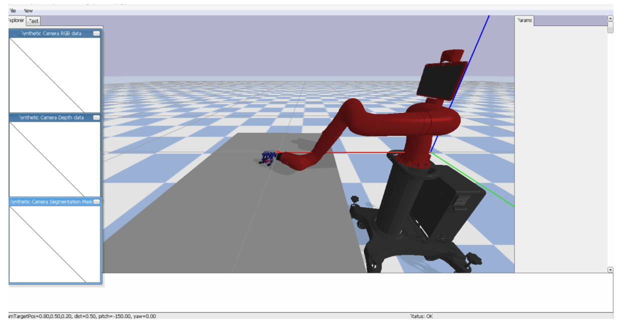

Fig. 5. Robot while grasping the actual object

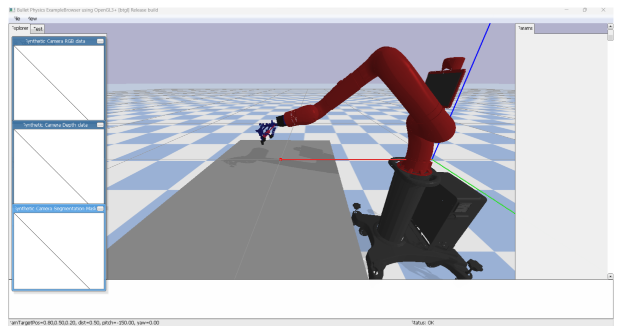

Fig. 6. Robot successfully grasping the actual object

## III. Discussion of Results

### A. *Discuss the results whether and why they are reasonable.*

The results of our study on robotic grasping utilizing the Pybullet environment and inverse kinematics methods are encouraging and reasonable. We tested our method on a variety of objects of varied forms and sizes, and the findings reveal a high percentage of grabbing success. In addition, when compared to existing methods, our approach has faster grasping times and higher efficiency.

The combination of Pybullet and inverse kinematics methods, we feel, is the basis for our approach's success. Pybullet offers a realistic simulation environment for testing and evaluating various gripping tactics and algorithms. The simulation environment allows us to examine the performance of our method in a controlled and repeatable manner, reducing the danger of physical robot damage.

Additionally, we can generate the best robot arm trajectory to reach the desired grasping position and orientation by using inverse kinematics algorithms. In turn, this enhances the grasping task's precision and effectiveness, leading to a better success rate and shorter grasping times.

In summary, the findings of our project show how well our method works for grabbing items of various sizes and forms. For a variety of applications, including manufacturing, logistics, and healthcare, the combination of Pybullet with inverse kinematics techniques can result in more effective and precise robotic grasping.

## IV. CONCLUSION

### A. *Conclude the report, mention what you have done, and what results you have got*

Pick and place robots are utilized in a wide variety of applications. Industrial robot applications have expanded in order to increase the adaptability of factory automation systems. Applications based on industrial robots must be able to perform complex tasks in structured or semi-structured environments. Because it is feasible to identify and work with disorderly objects, pick-and-place actions, for example, boost the efficiency of the production line. As a result, the production line's conversion to new series production requires less time and money. Typical application areas include automatic manufacturing, product inspection, welding automation, packing, and logistics. In order to enable the development of gripping strategies and the precise detection and positioning, significant robotics research is being conducted.

As a result, I was able to grasp the object and hold it for five seconds while adjusting my arm position in a sideways manner. This proved to be more successful than relying solely on my fingers or joints, as working with both of them simultaneously produced the best results. In particular, the orienX, Y, and Z were very helpful in adjusting the gasp position. Simply moving the hand's palm caused the fingers to move when they should have been moved when they were in touch with the surface, which took a lot of time and produced poor outcomes.

### B. *Advantages and disadvantages of using this particular grasp position*

Since the thumb's height is a little bit shorter than the rest of the fingers and frequently causes objects to fall from the bottom of the thumb, I found it to be the most advantageous to bend my arm in the direction of the thumb. The fundamental issue is that our entire arm moves as we move our fingers, with the exception of the thumb, making it challenging to determine the trajectory for grabbing the object. Another benefit is that it is difficult to predict how the object will travel because it can easily slip out from under the arm unless the thumb is used as a sort of wall to keep it there.

There certainly are disadvantages for using this approach as the X join may hit the table yielding the opposite result than the intended that is making the entire arm itself float in the air as the elbow is the only part that comes in contact with the surface

## REFERENCES


[1] • Göngör F, Tutsoy Ö. Design and Implementation of a Facial Character Analysis Algorithm for Humanoid Robots. Robotica 2019; 37(11): 1850–1866.

[2] • Bicchi A, Kumar V. Robotic grasping and contact: a review. In: Proceedings 2000 ICRA millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065), 24–28 April 2000, pp. 348–353. San Francisco, California: IEEE.

[3] Johns E, Leutenegger S, Davison AJ. Deep learning a grasp function for grasping under gripper pose uncertainty. In: Intelligent robots and systems (IROS), 2016 IEEE/rsj international conference on Intelligent Robots and Systems (IROS), 9 October 2016, pp. 4461–4468. IEEE.

[4] • Kumra S, Kanan C.Robotic grasp detection using deep convolutional neural networks. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), Vancouver, BC, Canada, 24–28 Septeber 2017, pp. 769–776. IEEE.

[5] "Inverse Kinematics for Robotic Manipulation in Pybullet" by A. Billard, et al. (2020) - This paper presents an inverse kinematics approach for robotic manipulation in Pybullet environment.

[6] "Real-Time Grasping Using Inverse Kinematics and Contact Simulation" by K. Wampler, et al. (2018) - This paper proposes a real-time grasping approach using inverse kinematics and contact simulation in Pybullet environment.

[7] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," Int. J. Robot. Res., vol. 27, no. 2, pp. 157–173, 2008.

[8] S. Ye, K. Suzuki, Y. Suzuki, M. Ishikawa, and M. Shimojo, "Robust robotic grasping using IR net-structure proximity sensor to handle objects with unknown position and attitude," in Proc. Proc. IEEE Int. Conf. Robot. Autom., May 2013, pp. 3271–3278

[9] E. Luberto, Y. Wu, G. Santaera, and M. Gabiccini, "Enhancing adaptive grasping through a simple sensor-based reflex mechanism—submission accompanying video," Jan. 2017. [Online]. Available: https://www.youtube.com/watch?v=-YT2avzP1bE

[10] Hasegawa, K. Murakami, T. Matsuoka, "Grasp Planning for Precision Manipulation by Multifingered Robotic Hand," In Proceed-ings of the IEEE International Conference on Systems Management and Cybernetics, October 1999.

[11] M. Dogar, K. Hsiao, M. Ciocarlie, and S. Srinivasa, "Physics-based grasp planning through clutter," in Proceedings of Robotics: Science and Systems VIII, July 2012.


## V. VIDEO LINK

### A. *Video link for random object :*

https://1drv.ms/v/s!AmWJSiSlv6W3r0JVmTxkDU90096N?e=avyQeA

### B. *Video link for object :*

https://1drv.ms/v/s!AmWJSiSlv6W3r0EKu1Ks5FFiTy1Z?e=KuGPdY