Life-science Fetch Robot for Dexterous Manipulation of Centrifuge Tubes

A project of the 2019 Robotics Course of the School of Information Science and Technology (SIST) of ShanghaiTech University

https://robotics.shanghaitech.edu.cn/teaching/robotics2019

Jiahui Zhu*, and Yizheng Zhang*

{*zhujh1, zhangyz*}@*shanghaitech.edu.cn*

Abstract

In the field of robotics, service robots are increasingly becoming an industry hot-spot. Especially for mobile manipulation, many researchers have studied its applications and some teams are focused on the manipulator's pose estimation of grasping objects, and the other teams are working on the simultaneous localization and mapping, path planning and automatic obstacle avoidance of mobile vehicles. Very few of them explore the integrated system of mobile manipulation and only transport simple, regular objects. Since in life science experiments, many repetitive, simple tasks can be replaced by autonomous robots. This work is aiming to apply a mobile manipulation Fetch robot to execute the dexterous manipulation of centrifuge tubes in life science laboratories. Usually, those tubes have a screw-lid so that operators need to unscrew the lid and add samples inside. In this work, we performed the experiments as follow: (a) to get an accurate centrifuge tube localization; (b) to grasp the tube from the top and place it on the base of Fetch; (c) to transport the tube from the base to the test table; (d) to unscrew the lid from the top of the centrifuge tube and drop it into a box.

I. Introduction

Thus life science is a research area that needs to do a lot of repetitive experiments. Usually, the only difference between experiments is the difference in parameters due to that there is a countless number of proteins, molecules and reagents that need to be explored by testing with them in real life science experiments. This is a work that requires manual labor that is quite often very repetitive and tiresome. Students in Universities or lab technicians in research institutes spend a lot of time on these experiments. Sometimes because of too tired, they make mistakes, and have to do it again. On the other hand, this kind of repetitive, simple experiment is perfect for robots to do it.

There is further literature regarding research on mobile manipulation for life science. [1] report on a robot called H20 that is concentrating on the hardware, control and planning aspects. Schmitt et al. [2] take robot dynamics and time-variant environments into consider and propose a new model for sequential manipulation tasks. It can interact with the object in the environment like human to avoid collision with them, which is pretty useful for assistant robots and life science robots. [3]describes a reinforcement learning (RL) strategy for manipulation and grasping of a mobile manipulator that reduces the complexity of the visual feedback and handle varying manipulation dynamics and uncertain external perturbations. [4] propose Neural Task Graph (NTG) Networks, which use conjugate task graph as the intermediate representation to modularize both the video demonstration and the derived policy. This network NTG improves data efficiency with visual input as well as achieve strong generalization without the need for dense hierarchical supervision

*These authors have contributed equally to this work.

In the work of [5], they use self-supervision to learn a compact and multimodal representation of sensory inputs like force sensor, which can then be used to improve the sample efficiency of our policy learning. Srinivasa et al. [6] present practical techniques that improve performance in areas like the house by considering the complete system in the context of this specific domain. In an industrial scenario Stoyanov et al. [7] build a nearly market-ready robot to unload coffee sacks out of sea freight containers in RobLog Project. They using feature-based object recognition, ROS and MoveIt! to solve those tasks successfully.

Fig. 1: A mobile manipulation Fetch robot in the life science laboratory.

II. State of the Art

A. JIAHUI

1) Papers: In the last few years, there are a few works done on mobile manipulation as an integrated system and several mobile manipulators have been coming to the market, for example, Tiago by Pal Robotics, Mobile Baxter by Rethink Robotics, and Fetch and Freight by Fetch Robotics [8].

For labware transportation in life science laboratories, a mobile robot called H20 with dual arms capable of 6-Degree of Freedom (DOF) mobility, an indoor GPS navigation system and Kinect sensor developed by H. Liu. et al. [9], [10] can realize manipulation tasks for different experimental objects such as flat panels and labware containers. However, different from general wheeled mobile manipulation, [11] proposes a lab automation drone and a robotic limb is attached to the robotic rotor-craft which could manipulate and transfer lines rapidly between test work-spaces.

2) One further paper relevant to the project: In the paper [12], they describe a method for robustly detecting visual fiducials and propose a graph-based image segmentation algorithm based on local gradients that allow lines to be precisely estimated. They specify and provide results on a set of benchmarks that will allow better comparisons of fiducial systems in the future. Their system is composed of two major components: the tag detector and the coding system.

- Detection process
  The detector attempts to find four-sided regions that have a darker interior than their exterior. Tags are composed of black and white blocks similar to the QR code and different tags contain different ID information. First, paste some tags in the scene. Secondly, the gradient at every pixel is obtained, and then compute their magnitudes and direction. Using a graph-based method, pixels with similar gradient directions and magnitude are clustered into components. Using weighted least squares, a line segment is then fit to the pixels in each component. Once four lines have been found, a candidate quad detection is created. Finally, they compute the 3×3 homography matrix that projects 2D points in homogeneous coordinates from the tag's coordinate system to the 2D image coordinate system. The homography is computed using the Direct Linear Transform (DLT) algorithm.
- Payload decoding
  The final task is to read the bits from the payload field. They do this by computing the tag-relative coordinates of each bit field, transforming them into image coordinates using the homography, and then thresholding the resulting pixels. In order to be robust to lighting (which can vary not only from tag to tag, but also within a tag), they use a spatially-varying threshold.

3) ROS package: AprilTag is a free and open source visual fiducial system developed by the APRIL Robotics Laboratory of the University of Michigan, useful for a wide variety of tasks including augmented reality, robotics, and camera calibration. Targets can be created from an ordinary printer, and the AprilTag detection software computes the precise 3D position, orientation, and identity of the tags relative to the camera. The fiducial design and coding system are based on a near-optimal lexicographic coding system, and the detection software is robust to lighting conditions and view angles. It uses a 2D bar code style "tag" similar to the QR code, allowing full 6-DOF localization of features from a single image.

In this work, we used an open source ROS package apriltag_ros for objects' pose estimation. The apriltag_ros package is a ROS wrapper of the AprilTag 3 visual fiducial detection algorithm. Provides full access to the core AprilTag 3 algorithm's customizations and makes the tag detection image and detected tags' poses available over ROS topics (including tf). The core AprilTag 3 algorithm is extended to allow the detection of tag bundles and a bundle calibration script is provided (bundle detection is more accurate than single tag detection). Continuous (camera image stream) and single image detector nodes are available. The package works as shown in Fig. 2.

Fig. 2: Overview of the apriltag_ros package.

B. YIZHENG

1) Papers: Schmitt et al. [2] propose a new model for sequential manipulation tasks that also considers robot dynamics and time-variant environments. It can automatically derive constraint-based controllers and use them as steering functions in a kinodynamic manipulation planner using this model. Thakar et al. [13] propose an active learning based approach that can pick-up parts while the mobile base and the gripper are moving to minimize the operation time.

Wang et al. [14] propose a generic framework for estimating the 6D pose of a set of known objects from RGB-D images and deploy the proposed method to a real robot to grasp and manipulate objects based on the estimated pose.

2) One further paper relevant to the project: In this work, the author present DenseFusion, a generic framework for estimating 6D pose of a set of known objects from RGB- D images. DenseFusion is a heterogeneous architecture that processes the two data sources individually and uses a novel dense fusion network to extract pixel-wise dense feature embedding, from which the pose is estimated. Furthermore, they integrate an end-to-end iterative pose refinement procedure that further improves the pose estimation while achieving near real-time inference. The experiments show that this method outperforms state-of-the-art approaches in two datasets, YCB-Video and LineMOD. The method's architecture contains two main stages. The first stage takes the color image as input and performs semantic segmentation for each known object category. Then, for each segmented object, feed the masked depth pixels (converted to 3D point cloud) as well as an image patch cropped by the bounding box of the mask to the second stage. The object pose is obtained by the second stage processes. It comprises four components:

- a fully convolutional network that processes.
- a PointNet-based network that processes each point in the masked 3D point cloud to a geometric feature embedding.
- a pixel-wise fusion network that combines both embeddings and outputs the estimation of the 6D pose of the object based on an unsupervised confidence scoring.
- an iterative self-refinement methodology to train the network in a curriculum learning manner and refine the estimation result iteratively.

We will try to use this algorithm to estimate the centrifuge tube's pose and then grasp it.

3) ROS package: The move_base package provides an implementation of an action that, given a goal in the world, will attempt to reach it with a mobile base. The move_base node links together a global and local planner to accomplish its global navigation task. It supports any global planner adhering to the nav_core::BaseGlobalPlanner interface specified in the nav_core package and any local planner adhering to the nav_core::BaseLocalPlanner interface specified in the nav_core package. The move_base node also maintains two costmaps, one for the global planner, and one for a local planner that are used to accomplish navigation tasks. The pip line of the package is shown in Fig. 3

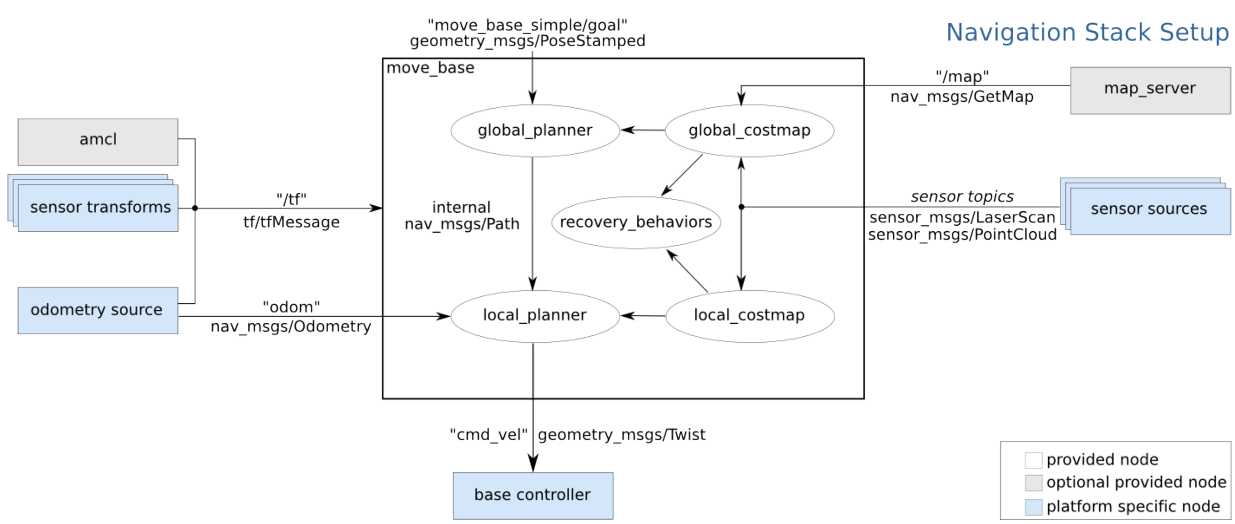


Fig. 3: The framework of the move base node.

C. *One more paper: Fetch robot*

In this project, we use the Fetch robot [15] which is a mobile manipulator consisting of a differential drive mobile base, an arm with 7 degrees of freedom and 6kg payload, a pan and tilt head, a torso lift actuator, and a standalone mobile robot platform, as shown in Fig.4. The mobile base includes a SICK laser scanner with a 220-degree field of view and a 25-meter range. Freight also includes a base-mounted 3D camera. Fetch includes a head-mounted depth camera. The gripper is a modularity point, allowing custom grippers to be swapped in, but supplied with a default parallel-jaw gripper capable of grasping a wide range of objects. Intel-based computers provide processing power for navigation, manipulation and perception activities, while extensive battery capacity gives each robot an 8- to 10-hour run time. Fetch and Freight Research Development software is based on ROS.

Since Fetch has a head-mounted RGB depth camera, we can use it to scan the Apriltag to locate the centrifuge tube. In this project, we need to mount a fixture on the base of Fetch robot to avoid the tube shaking during transportation, so we design a 3d printing connection part and make use of the base expansion mount points. Finally the fixture is mounted properly with two bolts.

Fig. 4: Overview of the Fetch robot.

III. System Description

The hardware of the experiment is shown in Fig. 5 and the system contains 6 components:

- MoveIt framework that is used to control the manipulator to grasp the object.
- Cartographer algorithm to build a map.
- Move base ros package that is used to navigate the robot to the target place where the object should be grasped.
- AprilTag 3 visual fiducial detection algorithm is used to obtain the object pose.
- We will design a device that can clamp for test tubes on the robot.

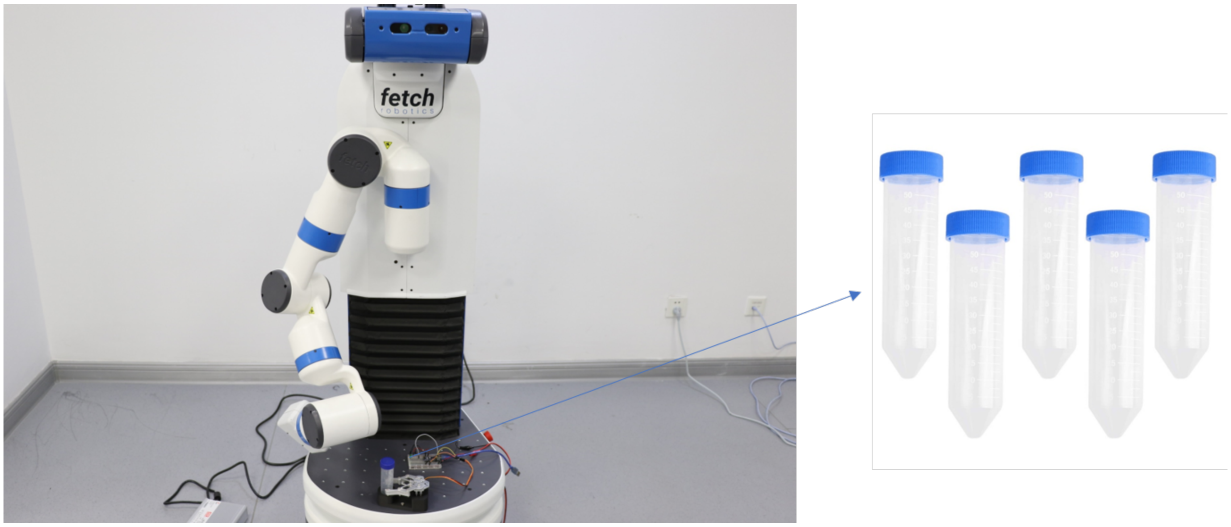


Fig. 5: Hardware of the experiment and centrifuge tubes with screw lids.

A. Actionlib software interface

Fetch Robot contains three actionlib services:

- gripper_controller/gripper_action: it can control the gripper open or close.
- gripper_controller/led_action: there are four led on the gripper, we can control the status of the led to represent the status of the robot.
- head_controller/point_head: it can control the head of the robot that decide where the robot look at.

B. Using MoveIt framework to control the manipulator to grasp the object

Fetch Robot contains three moveit groups:

- 7 DOF robot arm.
- 7 DOF robot arm with torso.
- Gripper.

The gripper group an alternative method to control the gripper, the more often used method is the actionlib method. For the most time, 7 DOF robot arm moveit group already can meet our needs to grasp the object. Torso is needed when we want the robot to grasp something high.

Firstly, we put the AprilTag on the test table. Once the camera sees the AprilTag, the pose of the tube is obtained and we set the coordinate as the goal of the inverse kinematics. The manipulator reaches directly above the tube, then drops vertically to grab it. Since the tube is put on a table, it is required to consider the collision between the robot arm and the table into account when the arm is planning the trajectory. So there is a collision object named ”table” being added to the moveit PlanningSceneInterface. See Fig. 7 left. After the tube is grasped, the manipulator places the tube on the fixture mounted on the robot, and then the robot can transport the tube to the desired location. Since we install the fixture on top of the robot base, it is required to consider the collision between the robot arm and the fixture into account when the arm is planning the trajectory. So there is a collision object named ”fixture” being added to the moveit PlanningSceneInterface. See Fig. 6 left.

After the arm is on top of the fixture, the only requirement of the arm is moving down. It only need one freedom movement, so we decide to move the torso, 7 DOF robot arm with torso moveit group is

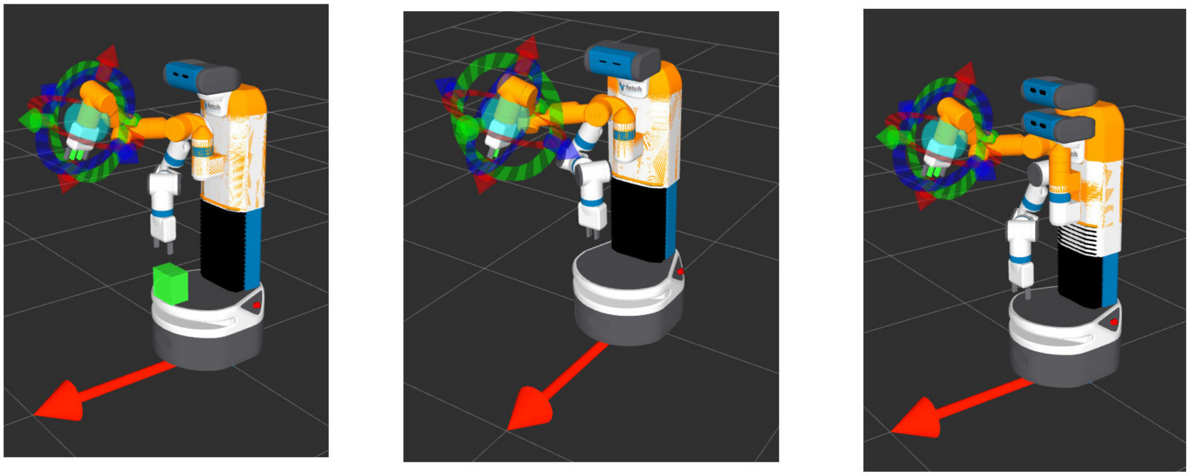


Fig. 6: The process of grasping the tube.

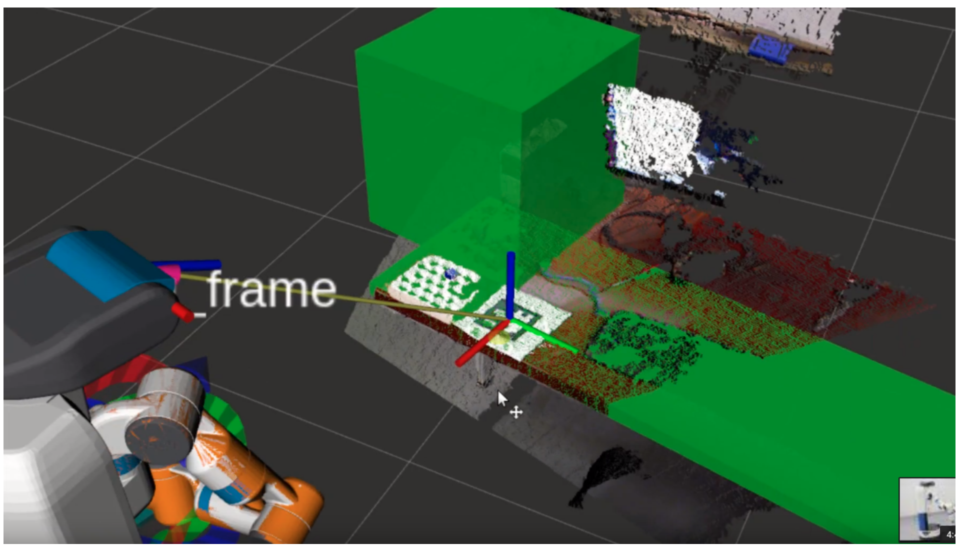


Fig. 7: Add the table as the collision.

used to do this work. On the other hand, the test tube attitude does not allow rotation, so it has to add the orientation constrain when the arm is planning. We use moveit_msgs::Constraints to add the constraints to the moveit. The quaternion is set as (x:0.707, y:0, z:-0.707, w:0) to keep the gripper face down. See Fig. 6 middle.

Due to the error of the robot arm, it will lead to inaccurate tube grasping. Fortunately, the fixture we designed can mechanically correct this error. After the robot reach the desired location, it sees the AprilTag again, and obtains the goal position where the tube should be placed. If the robot cannot see the AprilTag, it will turn its head to find the AprilTag. Once seeing the AprilTag, it can adjust its body so that it is facing the AprilTag. The robot will pick up the tube placed on the fixture, and plan the path directly above the goal position, and then drop it vertically to insert the tube into the goal position. Then, it unscrew the lid from the top of the centrifuge tube (Fig. 5) and place the lid to a box. Now we can add something into the tube if needed.

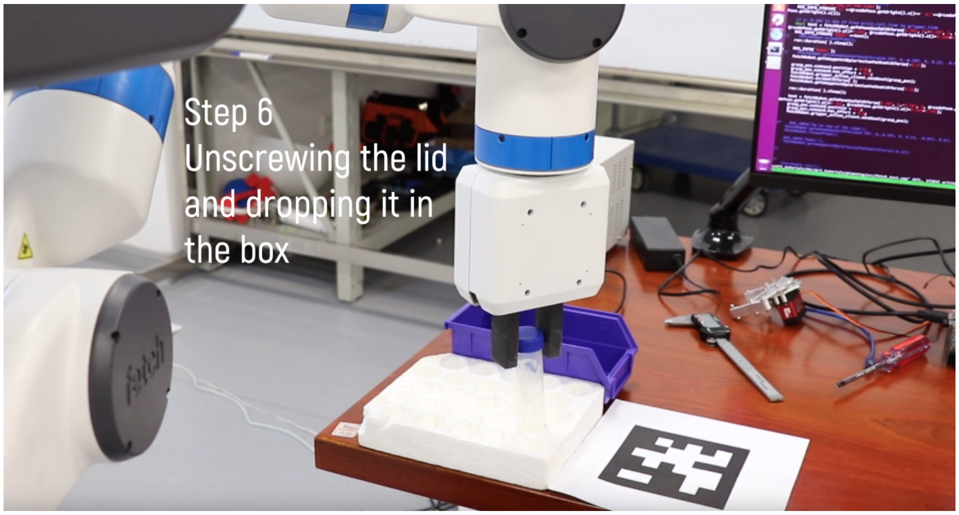


Fig. 8: Unscrew the lid.

C. Unscrew the lid

After the arm grasp the tube, the robot can rotate the last joint to unscrew the lid. It also needs moveit to finish. The trajectory is planned in the joint space. A sequence of the desired angle is sent to make the joint rotate. But maybe due to the framework bug, in this part, the robot can not plan successfully. See Fig. 8

D. Fixture design of the centrifuge tube

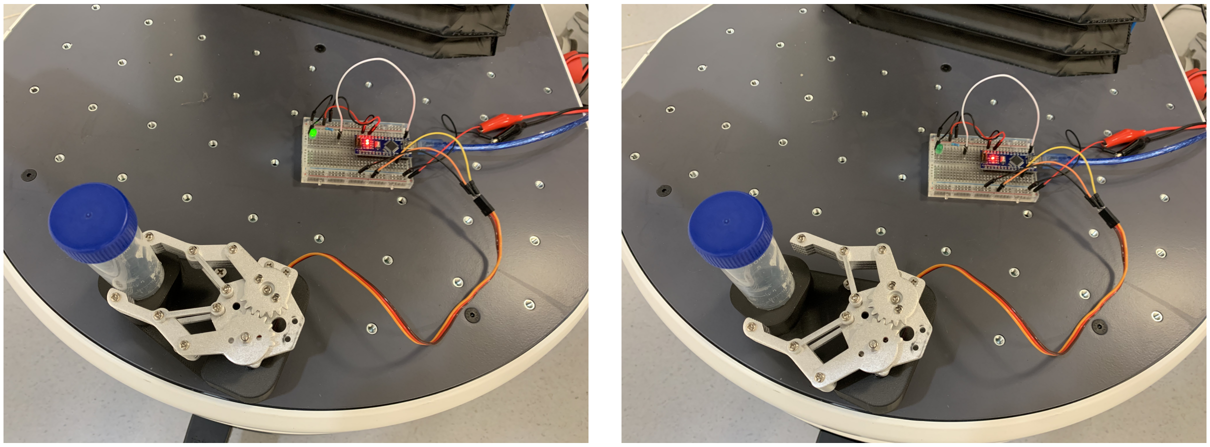
The fixture of the centrifuge tube contains two parts, a clamping part, and a fixed part. The gripper is controlled by the servo to clamp and release, as shown in Fig.9 and the servo library of arduino can be used to easily control the stroke and speed of the servo. Considering that the gripper and the centrifuge tube need to be mounted on the platform of Fetch, we used SolidWorks 3D modeling software to design a base that can hold the centrifuge tube and the gripper, and then we printed it with a 3D printer. Finally, we bolted the entire fixture to the mobile robot platform.

E. Communicate between Fetch and the fixture using rosseiral ROS package

Rosserial is a protocol for wrapping standard ROS serialized messages and multiplexing multiple topics and services over a character device such as a serial port or network socket. It contains several client libraries allowing users to easily get ROS nodes up and running on various systems. In our case, we use a arduino board to control the fixture, the rosserial_arduino is used in our case. There is a subscribe node running in arduino which subscribe the command from the robot. It is a trigger information so an empty message already meets our needs. When the robot arm picks the tube on top of the fixture, the robot sends an empty message to trigger the fixture to fix the tube.

F. Tricks in the project

- Measure the relative position between AprilTag and tube: write the relative position as the ros param and load it from rosparam server can be more convenient than write it in the cpp code. If write it in the cpp code, every time the param changed, it has to be compiled again, which is unconvenient.



(a) Gripper clamping (b) Gripper releasing

Fig. 9: The fixture and the arduino board are mounted on the base of Fetch.

- Limit the acceleration of the motor: it helps robot moves smoother.
- After the robot arm reaches the target point, pause for a while to prevent the robot body from shaking and causing errors.
- If the manipulator planner can not plan successfully frequently, consider change another planner.

IV. System Evaluation

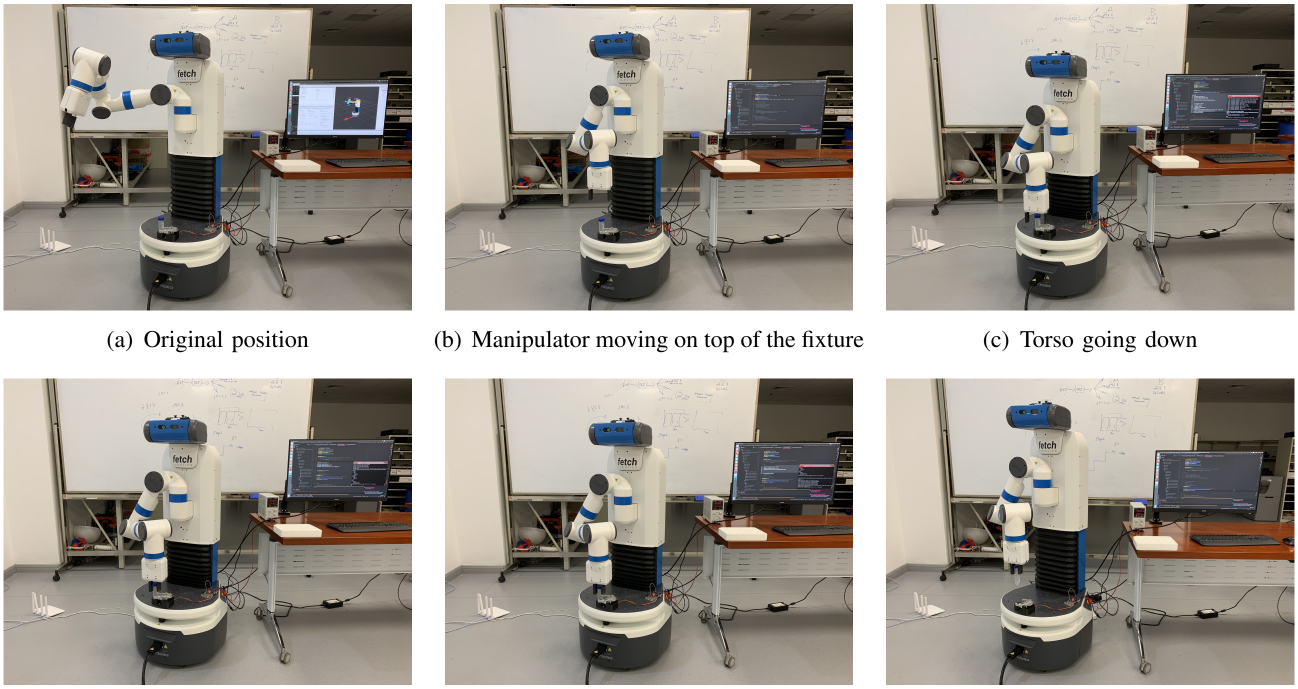
This work aims to use the Fetch robot for grasping and placing the centrifuge tube where the desired labware is positioned. Due to the centrifuge tube has a lid to prevent the sample leaked, we need to use the Fetch robot's parallel gripper to unscrew the lid before the sample is added into the tube. At first, we conducted several experiments to prove that it is possible to plan the manipulator to grasp the tube from the base, as shown in Fig.10. The demo of whole experiments can be seen at the YouTube video.

We choose two criteria to evaluate the performance.

- Success rate of grasping and placing
  After the robot moves to the designated position, it needs to grasp the tube and place it on its fixed platform. We can do it in 50 attempts and see how many times it successfully grasps and places the tube. Through the experiment, we found that the success rate was 70% to 80%.
- Alignment error
  Because the centrifuge tube need to be placed on the fixture and then transports to the desired position. At this point, the alignment error between the tube and the fixture is involved. We can evaluate the performance of the manipulator by measuring the eccentricity distance between the tube body and the fixture. We found that the alignment error was around 1 cm which didn't effect the process because the tube has a conical bottom and this helped the test tube insert into the fixture.

V. Conclusions

In this project, we have proposed a method for labware transportation and dexterous manipulation of centrifuge tubes in life science laboratories using a Fetch robot. In the future, the development of repeatable and simple experiments in life science laboratories, very often plagued with manual transportation, could profit from this.



(a) Original position (b) Manipulator moving on top of the fixture (c) Torso going down

(d) Parallel-jaw gripper grasping the tube (e) Fixture releasing (f) Torso lifting up

Fig. 10: Steps of grasping the centrifuge tube.

VI. How to Reproduce the Project

Repo: Robotics2019/robotics2019_project_LiferRobot

```
cd catkin_indigo_ws
catkin_make
roslaunch fetch_moveit_config move_group.launch
rosrun rosserial_python serial_node.py /dev/ttyUSB0
roslaunch apriltags2_ros continuous_detection.launch
roslaunch life_robot_moveit start.launch move_real_robot:=true
```

VII. Appendix

A video demo of our project can be found at https://youtu.be/GrqMmCn_Z8M .

References

[1] M. M. Ali, H. Liu, N. Stoll, and K. Thurow, "Intelligent arm manipulation system in life science labs using h20 mobile robot and kinect sensor," in *2016 IEEE 8th International Conference on Intelligent Systems (IS)*, pp. 382–387, IEEE, 2016.

[2] P. S. Schmitt, F. Wirnshofer, K. M. Wurm, G. v. Wichert, and W. Burgard, "Modeling and planning manipulation in dynamic environments," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 176–182, May 2019.

[3] Z. Li, T. Zhao, F. Chen, Y. Hu, C. Su, and T. Fukuda, "Reinforcement learning of manipulation and grasping using dynamical movement primitives for a humanoidlike mobile manipulator," *IEEE/ASME Transactions on Mechatronics*, vol. 23, pp. 121–131, Feb 2018.

[4] D.-A. Huang, S. Nair, D. Xu, Y. Zhu, A. Garg, L. Fei-Fei, S. Savarese, and J. C. Niebles, "Neural task graphs: Generalizing to unseen tasks from a single video demonstration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8565–8574, 2019.

[5] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," *arXiv preprint arXiv:1810.10191*, 2018.

[6] S. S. Srinivasa, A. M. Johnson, G. Lee, M. C. Koval, S. Choudhury, J. E. King, C. M. Dellin, M. Harding, D. T. Butterworth, P. Velagapudi, *et al.*, "A system for multi-step mobile manipulation: Architecture, algorithms, and experiments," in *International Symposium on Experimental Robotics*, pp. 254–265, Springer, 2016.

[7] T. Stoyanov, N. Vaskevicius, C. A. Mueller, T. Fromm, R. Krug, V. Tincani, R. Mojtahedzadeh, S. Kunaschk, R. M. Ernits, D. R. Canelhas, *et al.*, "No more heavy lifting: Robotic solutions to the container unloading problem," *IEEE Robotics & Automation Magazine*, vol. 23, no. 4, pp. 94–106, 2016.

[8] M. A. Roa, D. Berenson, and W. Huang, "Mobile manipulation: Toward smart manufacturing [tc spotlight]," *IEEE Robotics & Automation Magazine*, vol. 22, no. 4, pp. 14–15, 2015.

[9] H. Liu, N. Stoll, S. Junginger, and K. Thurow, "Mobile robot for life science automation," *International Journal of Advanced Robotic Systems*, vol. 10, no. 7, p. 288, 2013.

[10] M. M. Ali, H. Liu, N. Stoll, and K. Thurow, "Kinematic analysis of 6-dof arms for h20 mobile robots and labware manipulation for transportation in life science labs," *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. 10, 2016.

[11] D. Kim and P. Y. Oh, "Lab automation drones for mobile manipulation in high throughput systems," in *2018 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–5, IEEE, 2018.

[12] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3400–3407, IEEE, 2011.

[13] S. Thakar, P. Rajendran, V. Annem, A. Kabir, and S. Gupta, "Accounting for part pose estimation uncertainties during trajectory generation for part pick-up using mobile manipulators," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 1329–1336, May 2019.

[14] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[15] M. Wise, M. Ferguson, D. King, E. Diehr, and D. Dymesich, "Fetch and freight: Standard platforms for service robot applications," in *Workshop on autonomous mobile service robots*, 2016.