Object Manipulation in RoboCup@Home

Gwendal Paton

Master SIIA

6/3/2023 - 21/7/2023

Supervisor :
Cédric Buche
buche@enib.fr

Abstract

This report presents my internship in IRL CROSSING and CERV from March to July 2023. The goal of this internship was to find a solution to make a Pepper robot from United Robotics Group grasp objects. This work was then used in the RoboCup@Home competition which we won. In this report, I present the RoboCup@Home competition and my involvement with team RoboBreizh as well as explaining my contribution to grasping. In the end grasping was a difficult task due to how Pepper is built.

Keywords : Robotics, ROS, Manipulation, Grasping, RoboCup@Home

Acknowledgments

First I would like to thank Cédric BUCHE for this great opportunity and for his supervision. Then I would like to thank all of the RoboBreizh Team : Thomas Ung, Maëlic Neau, Louis Li, Sinuo Wang, Cédric Le Bono, Duc Nhan Do, Rebecca and Samuel who all did outstanding work on the project. I also want to thank all the people working at CROSSING. Finally I would like to thank the other interns at CERV and IMT Atlantique.

Contents

Acknowledgments **2**

Introduction **4**

1 The RoboCup@Home Competition **5**
1.1 Overview . . . 5
1.2 Qualification . . . 6
1.3 Environment . . . 7
1.4 Our robot's architecture . . . 8
1.5 Schedule & Awards . . . 9
1.6 Tasks & Scoring . . . 9
1.6.1 Stage 1 . . . 10
1.6.2 Stage 2 . . . 12

2 Contribution : Grasping and robot movement **14**
2.1 Goal and state of the arts . . . 14
2.1.1 Point cloud segmentation . . . 14
2.1.2 Grasping Algorithms . . . 14
2.1.3 MoveIt . . . 15
2.1.4 Problems encountered . . . 17
2.2 Implemented Solution . . . 18

3 Contribution : Other work **19**

4 Result **20**
4.1 Problems encountered during the competition . . . 20

Conclusion **22**

Introduction

The main goal of this internship was a participation in the Robocup Competition with team RoboBreizh. Our team is participating in the Robocup@Home league, in the Social Standard Platform League (SSPL). The robot used in SSPL is United Robotics Group's Pepper (previously owned by Aldebaran Robotics and Softbank Robotics). Team RoboBreizh won the previous edition in Bangkok, Thailand (1), so we had some solid background for this year's edition. Team RoboBreizh also won the RoboCup@Home Education 2020 Best Performance award (2). One of the main things to improve was object manipulation and grasping, which was my task.

This internship lasted 5 months, from the 6th of March 2023 to the 21st of July 2023. The majority of team members were working at CROSSING : an International Research Laboratory (IRL) from CNRS (Centre national de la recherche scientifique) based in Adelaide, Australia. I spent 3 months in Adelaide with the team to get some base knowledge on the robot and plan our strategy for the competition, then I returned to France, at the Centre Européen de Réalité Virtuelle (CERV) in Plouzané to finish my work.

The competition took place in Bordeaux, France from the 4th to the 10th of July 2023. During the competition I helped on robot movement related issues and on a lot of other various tasks. We spent the whole competition working non stop to make sure we would win. After the competition, I updated all the documentation and fixed some issues that were uncovered during the competition.

In this report I will first explain what the RoboCup@Home competition is, then I will describe my contribution to the project and reflect on the work done.

1 The RoboCup@Home Competition

1.1 Overview

RoboCup is a widely recognized initiative that provides a space for discovering and sharing new advancements in robotics and AI. It consists of different leagues, each focusing on specific areas. The competition aims to push the limits of what robots can do.

Our league is the RoboCup@Home, which is dedicated to the development of robots that help people in household environments, performing tasks that enhance the quality of daily life. These tasks include navigation through complex indoor spaces, recognizing and manipulating objects, communicating with humans, and planning actions. By focusing on natural human-robot interaction and encouraging innovation in areas such as communication, vision, navigation and manipulation, RoboCup@Home brings us closer to a future where robots are an integral part of our homes. Whereas some other leagues look more like a traditional sport, our league is more of a showcase of what our robot is capable of.

The RoboCup competition features various other leagues, each focusing on specific challenges. The RoboCup Soccer leagues, for instance, makes teams design autonomous soccer-playing robots that have to display teamwork, strategy, and agility like true soccer players to score goals. Other leagues are focusing on areas such as rescue operations, where robots are tasked with simulating disaster scenarios to demonstrate their potential for saving humans and assisting rescuers.

RoboCup is a hub of innovation, a stage where researchers and teams showcase new ideas and advancements. It is an international event so researcher from all around the world will meet and discuss ideas Robotics and all subjects related to it like AI are growing fast, so these kind of event are good to keep everyone up to date and discuss new evolution. It also engages students, teachers, and the public, creating interest in science, technology, engineering, and robotics. With this kind of healthy competition, people are more motivated to pursue research in robotics.

1.2 Qualification

To participate in RoboCup we had to register and qualify. I did not help for this part, but given that we were previous year winners we already had the majority of things needed to qualify. All rules relative to RoboCup@Home are found in the rulebook (3). The first step to qualify is registration. This involves providing information about the team, the robot that will be used, and what technology are used. Teams have to submit technical documentation detailing their robot's hardware, software architecture, algorithms, and approaches to solving the various challenges outlined by the competition. In our league, everyone has the same robot : Pepper, so we just detailed all software and models onboard. The submitted paper is called the team description paper.

The organizers review this paper and evaluate it based on predetermined criteria, including the robot's ability to navigate, recognize objects, interact with humans, and manipulate objects, etc... After the evaluation of the technical documentation, the organizing committee of RoboCup@Home selects a certain number of teams to participate in the competition. The selected teams are then officially invited to the competition.

Teams are also required to provide a website with information about the team members and a summary of relevant contributions. Lastly all teams must provide a qualification video. In this video the robot is supposed to demonstrate clearly what was written in the team description paper, like navigation, object recognition, etc...

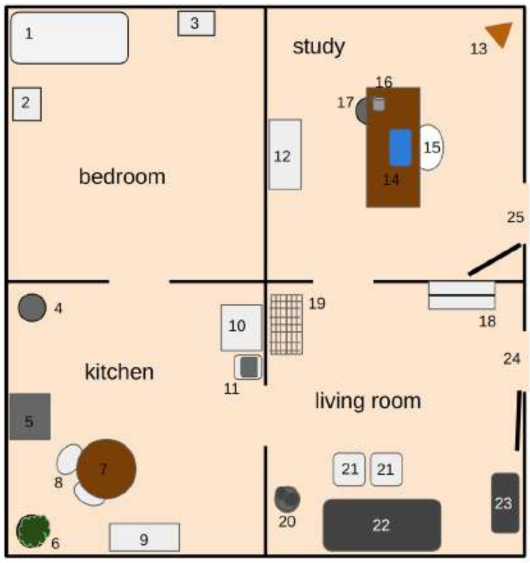

Figure 1: RoboCup@Home arena, representing an apartment with 4 rooms

1.3 Environment

The arena for the RoboCup@Home competition is designed like a domestic environment, it is supposed to represent an apartment with 4 rooms : bedroom, kitchen, living room and study (Fig. 1). The competition takes place indoors to simulate real-world home settings. The arena includes features like walls, furniture, doors, and objects commonly found in a household. Walls are not as tall as real walls, they are only 1 meter tall allowing people to see what's going on inside during the tasks. Many objects from the YCB dataset are scattered around the arena. Some other object that are not part of any known dataset are also added, so that team have to train new object recognition algorithms during the competition. One crucial part of RoboCup@Home is to map the whole arena, so that our robot can use this map to navigate. All teams take all possible measurements to have the best map possible.

1.4 Our robot's architecture

Our robot works with ROS. Pepper is not compatible with ROS from the get go, so we had to put another OS called Gentoo on top of the regular Pepper OS to make it work.

ROS (4), which stands for Robot Operating System, is an open-source framework that provides a collection of software tools and libraries to help developers build and control robots. Despite its name, ROS is not a traditional operating system like Windows or Linux. It is a middleware layer that sits on top of an existing operating system and provides a set of tools and services to help robot software development. Ros supports the Python and C++ language. During my internship I mainly used Python, but I had to use some C++ when using some algorithms.

ROS is designed to simplify the development process of robotics applications by offering a common modular architecture enabling developers to create and integrate software components (nodes) that can communicate with each other through asynchronous communication mechanisms. Two key concepts in ROS are topics and messages.

A ROS topic is a channel for communication between different nodes in a ROS system. Nodes can publish data to a topic or subscribe to a topic to receive data. Topics facilitate the exchange of information in a publish-subscribe pattern, where one node publishes data to a topic, and any number of other nodes interested in that data can subscribe to the topic and receive updates.

ROS messages are the data structures that nodes use to communicate with each other via topics. Messages define the structure and format of the data being exchanged. They can represent various types of information, such as sensor readings, motor commands, or any other relevant data. Messages are defined using a simple text-based format, and they are used to specify the content and structure of the data being sent or received on a topic.

Our architecture is based on 5 main packages. The main package is manager_pepper which controls everything. It oversees the robots behaviour. Tasks are organised with Petri Net Plans which allows us to make finite state machines to control our robot's behaviour. The other packages are navigation_pepper, which manages navigation and mapping, perception_pepper which focuses on image recognition models and computer vision, dialog_pepper which manages natural language perception, text to speech and speech to text, and manipulation_pepper which manages the robot movement and all task related to grasping and manipulation. I mainly worked on the manipulation_pepper package.

1.5 Schedule & Awards

The competition is 6 days long. It begins with 2 days of setup where teams get or unpack their robots and then begin preparing for the competition. All teams must provide a poster to present their work on the robot. This poster is then presented to a jury by a team member and some points are granted based on the poster and the presentation quality.

Then, all teams have to go through a robot inspection conducted by a referee to ensure the robot's operational safety, like preventing collisions with humans and having a emergency shutdown button.

Once the inspection is finished and validated, teams proceed to the first stage. In the first and second stages the robot must execute various tasks. Details about what the robot should do in each task is found in the RoboCup@Home Rulebook (3). After stage 1, the top 6 teams (or top 50% depending on the number of participants) advance to the second stage, which introduces more demanding tasks, often expanded versions of those encountered in the initial stage.

Lastly, the final round features the top 2 teams in an open showcase. After all this, winners are announced and prices are given.

1.6 Tasks & Scoring

For the tasks, points are awarded by a referee that ensures that the robot is performing the demanded tasks correctly. All the teams know the tasks beforehand, only small details such as the arena layout or objects used are given during the actual competition. Tasks are divided in 2 stages, with stage 2 tasks being harder. You must make it to the top half scores of stage 1 to be allowed to continue to stage 2.

This year there were 5 Tasks in stage 1 named Receptionist, Storing Groceries, Carry my Luggage, General Purpose Service Robot (GPSR) and Serving Breakfast. Stage 2 tasks are Clean the Table, Restaurant, Stickler for the Rules and Enhanced Global Purpose Service Robot (EGPSR). After stage 2 the top 2 teams advance to the finals.

Unlike the earlier stages where specific tasks are assigned, the finals provide teams with a more open format. Teams are given the flexibility to design and execute their own scenarios, tasks, and interactions to highlight unique features and innovations of their robots. The finals are typically judged by a panel of experts in robotics and AI. Evaluation criteria can include the novelty of presented features, the overall performance of the robots, the creativity of the scenarios, and the robots' interaction with the audience and judges.

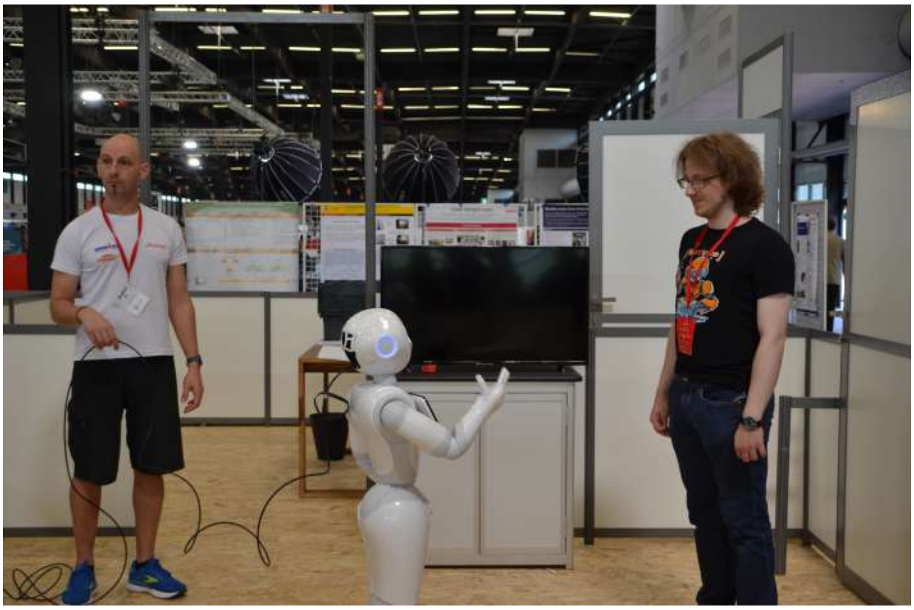

Figure 2: Our Pepper performing the Receptionist task

1.6.1 Stage 1

Receptionist

In the Receptionist task, the robot's objective is to welcome and guide two new guests to the living room, introducing them and ensuring they have a place to sit. The task emphasizes human-robot interaction, person detection, and recognition. The setup occurs in the living room, where the robot starts near the entrance door. The host and guests, each with names and favorite drinks, are part of the scenario. The robot must clearly introduce each guest as seen in figure 2, stating their name and favorite drink, and point out available seating. Additionally, describing each guest's characteristics earns bonus points, and the robot's gaze direction during interactions and navigation also awards points.

Storing Groceries

In the Storing Groceries task, the robot's primary objective is to transfer five objects from a table to a cabinet while arranging them based on categories or similarity. Optional goals includes opening the cabinet door, handling a tiny object, and managing a heavy object. The task focuses on object recognition and manipulation. The setup involves the robot starting outside the Arena and moving to the designated testing area upon door opening. The area features a cabinet and a table with objects placed on it, ready for the robot to grasp. The cabinet contains objects sorted into groups on different shelves, and the cabinet door may be open by default or closed for additional points. The robot must efficiently transfer objects from the table to the cabinet, potentially dealing with cabinet doors and various object sizes.

Carry my Luggage

In the Carry my Luggage task, the robot's task is to assist the operator by carrying luggage to a car parked outside. The primary goal is to transport a bag to the car, with optional objectives including re-entering the arena and avoiding people and obstacles. The challenge focuses on person following, navigation in new environments, and social navigation. The setup involves predefined starting locations, and the robot must pick up the designated bag, inform the operator that it is ready to follow them and navigate through obstacles.

General Purpose Service Robot (GPSR)

In the General Purpose Service Robot task, the robot is directed to execute arbitrary commands from an operator. The robot's objective is to execute three commands provided by the operator. This task focuses on intent recognition and task planning. The setup involves the task occurring inside the Arena, with potential commands requiring the robot to venture outside. The robot starts outside the Arena, moving towards an Instruction Point upon door opening. A Professional Operator (referee) or Non-Expert Operator (spectator) speaks out commands, which the robot executes. Command are generated with a standard generator but are so diverse that it's impossible to prepare for all cases without using some kind of AI model. The robot then returns to the Instruction Point for the next command. Points are given when the command is understood, and more points are given if the command is executed as well.

Serve Breakfast

In the Serve Breakfast task, the robot is required to set up a breakfast table for one person and prepare cereal for them. The primary goal is to put breakfast items (bowl, spoon, cereal box, and milk carton) on a table and preparing cereal. The task focuses on object perception and manipulation. The setup involves the robot starting outside the Arena and moving to the kitchen when the door opens. The test takes place in the kitchen, where no human presence is required unless the robot encounters difficulties. The robot serves breakfast on a chosen table, with chairs possibly present around it. Objects are prepositioned, and the robot's actions are structured, from selecting the table surface to placing items and pouring cereal and milk. Optional actions involve pouring milk into the bowl and strategically placing the spoon next to it.

1.6.2 Stage 2

Clean the Table

In the Clean the Table task, the robot's main objective is to clear a dining table by putting all dishes and cutlery in a dishwasher. This involves placing five items inside the dishwasher. Optional goals include opening the dishwasher door, pulling out the dishwasher rack, correctly placing items inside the dishwasher, and inserting a dishwasher tab. The task focuses on object perception and manipulation. The setup takes place in the kitchen, with a dining table and a plastic tray nearby. The table contains five objects representing a typical meal setting for one person, including silverware and tableware. The robot must effectively clean the table by transferring the items into the dishwasher.

Restaurant

In the Restaurant task, the robot is tasked with retrieving and serving orders to multiple customers in an actual, unfamiliar restaurant setting. The main goal involves detecting customers calling or waving, autonomously reaching their tables without prior guidance, and taking and serving their orders. This task focuses on task planning, mapping, navigation in unfamiliar environments, gesture detection, verbal interaction, and object manipulation. The setup takes place in a real restaurant or a suitable location other than the Arena. The robot starts near near a Professional Barman (or referee) to fulfill orders. The restaurant may have real customers, waiters, and different table configurations. Edible and drinkable objects from the standard object list constitute the orders, and the robot's actions encompass taking orders, placing them, and delivering to customers' tables.

Stickler for the Rules

In the Stickler for the Rules task, the robot's role is to ensure that house rules are adhered to during a party scenario. The primary goal involves identifying party guests who are breaking house rules, politely clarifying the rules to them, and confirming their compliance. The task focuses on object perception, human perception, action recognition, and verbal interaction. The setup takes place within the Arena, with the robot starting in the living room. The rules are :

- Every guest must have a drink in their hands
- No guest should be wearing shoes
- No guest should be in the bedroom
- There should be no litter on the floor

Enhanced Global Purpose Service Robot (EGPSR)

The "Enhanced General Purpose Service Robot" task is almost the same thing as GPSR. Commands generated are harder for the robot to understand and execute.

2 Contribution : Grasping and robot movement

2.1 Goal and state of the arts

One of the main challenges of Robocup@Home is object manipulation. In our league, the Social Standard Platform League (SSPL), it is very difficult for our robot, Pepper, to grasp any object as his arms and hands are not really meant for that purpose.

The first solution considered was to use a point cloud segmentation algorithm on an object, feed its point cloud into a grasping algorithm to get the point in which the robot must grasp the object, and then give this point as a target for MoveIt (5), which will then move the robot in the correct position.

2.1.1 Point cloud segmentation

The point cloud segmentation was done by another intern at IRL CROSSING. Using our object detection models, we get the bounding box coordinates of the object we want to grasp. Using this, we slice the point cloud given by the depth camera accordingly and are left with the entire point cloud of the object.

2.1.2 Grasping Algorithms

Our objective is to do efficient object grasping in the context of the RoboCup@Home competition. The majority of items that we will have to grasp are taken from the YCB dataset (6). The chosen algorithm should show great result on the YCB-Video dataset. The algorithm should be lightweight and robust considering the robot's sensors and limited computing power. Several grasping algorithms were considered, with S4G (7) demonstrating speed and REGNet (8) offering better overhaul accuracy. For previous years, a similar solution was tested using the Grasp Pose Detection (GPD) algorithm (9) (10). GPD was trained on the YCB Video dataset which is what we were looking for, and it also has a ROS wrapper. Using all this we could send a point cloud message to a ROS service, then GPD would give us a set of positions where we could grasp the object.

2.1.3 MoveIt

MoveIt (5) is a widely used open-source motion planning framework designed for the Robot Operating System (ROS). It offers a comprehensive set of tools and libraries that enable robots to efficiently plan and execute motions while avoiding obstacles. By providing motion planning algorithms, kinematics calculations and collision checking mechanisms, MoveIt simplifies the process of developing robotic applications. It also supports visualization, grasp planning, and manipulation tasks, enhancing the capabilities of robots in various environments. Integrated with ROS, MoveIt serves as a useful resource for robotics research and industrial applications, enabling developers to create collision-free trajectories and control robotic systems easily.

For MoveIt to work we need to configure it to our robot. For that we need a 3D model of our robot as well as a description of all mechanical joints between all of the robots' parts. For example there is a joint between Pepper's head and torso allowing 2 rotations. MoveIt allows us to easily simulate the robots movement in a Gazebo or Rviz environment. Gazebo and Rviz are an open-source simulation platform with a strong connection to ROS. They serves as a tool for simulating robots in a 3D environment where users can design and simulate scenarios. Gazebo and Rviz can simulate ROS-based robots and sensors, allowing developers to test their algorithms in a realistic virtual environment before deploying them onto actual hardware. The MoveIt interface in Rviz can be seen in Figure 3. There is a tool to automatically setup a MoveIt configuration with a URDF file as input. I did this configuration several times with different parameters in hope of finding the best outcome.

Once the MoveIt configuration is generated, it provides a ROS launch file that we can run, we can now access the robot's current position through MoveIt and give movement instruction directly from Gazebo. We can use MoveIt in python through a wrapper. We have to create a "Move Group Interface" which allows us to interact with a "move group". A move group is a set of joints describing a robot's part, for example all the joints in the right arm. We can then give a target position to reach. It will then plan the movement using its planner and execute the movement.

MoveIt uses several mechanical solvers to find the correct movement trajectory to execute. Our robot has a very limited computing power, so the chosen solver should be the fastest. We don't want to lose a lot of time calculating trajectories because during the competition we have a very limited time to execute all tasks. A study by Liu et Liu (11) did a complete benchmark of the solvers used in MoveIt. The results can be found in Figure 4. From that study, it appears that the RTTConnect solver is significantly faster than the other solvers while maintaining high accuracy. Given this result I used the RTTConnect solver while using MoveIt as it was the best solution for our goal.

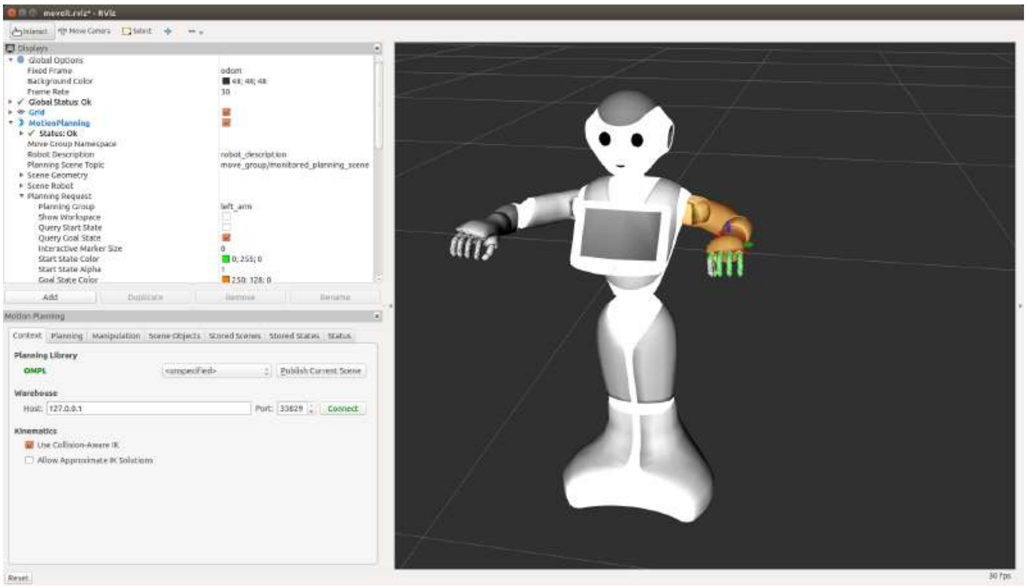

Figure 3: MoveIt interface with Pepper in Rviz

Planner	Solve (%)	Path time (s)	Path smoothness	Path clearance	Total time (s)
PRM	100	0.87	0.29	0.046	0.89
RRT	26	0.98	0.32	0.051	10.0
LazyPRM	46	0.60	0.30	0.070	10.0
RRTConnect	100	0.064	0.34	0.044	0.068
CHOMP	0	-	-	-	-
STOMP	94	0.17	0.051	0.163	0.19
PRM-CHOMP	64	0.99	0.050	0.046	1.70
RRT-CHOMP	16	0.26	0.070	0.053	10.0
LazyPRM-CHOMP	32	0.54	0.050	0.049	10.0
RRTConnect-CHOMP	64	0.175	0.070	0.045	0.47
PRM-STOMP	92	0.74	0.083	0.158	0.83
RRT-STOMP	28	0.70	0.077	0.128	10.0
LazyPRM-STOMP	42	0.98	0.072	0.140	10.0
RRTConnect-STOMP	96	0.12	0.075	0.160	0.16

Figure 4: MoveIt solver benchmark in industrial setting, from Liu et Liu (11), Table 2

2.1.4 Problems encountered

In theory, hooking a grasping algorithm to MoveIt and having Pepper grasp the object at the designated point should work, however while implementing this multiple issues appeared. Pepper's arms were not designed to grasp objects. Pepper has been advertised as a social robot, which is reflected in the name of our league : the Social Standard league. Its design was made to make it easier for humans to interact with it. While Pepper does have arms with limited mobility and basic grippers, its grasping abilities are not as advanced or precise as those of robots specifically designed for tasks that require complex object manipulation. Pepper's arms are more suited for basic gestures and interactions, such as waving, pointing, or touching, rather than intricate grasping of objects. Pepper's arms are also really short. It is nearly impossible to take an object on the floor as Pepper can't reach it. The same goes for objects too high, in a cabinet for example.

Another issue is that when given the same movement command, Pepper might inaccurately execute it. I tried simply raising an arm by commanding the arm pitch joint, and from one try to another results were not exactly the same. Sometimes the arm would stop mid-way for no reason. So even when giving proper commands, we are not sure that the robot would execute them properly. Another problem when grasping is the gripper, in our case Pepper's hand. To make it more acceptable and to facilitate more natural and intuitive interactions between Pepper and humans, it has a human-like hand. However it has poor strength and we can not control one finger at a time. Most gripping algorithms are designed for parallel grippers, which are the most common for robotic object manipulation. I tried to approximate Pepper's hand geometry as a parallel gripper to make it work. Grasping algorithms using a humanoid hand exist but are far less accessible and efficient as those for parallel grippers, and the inability to control one finger at a time would have made it useless nonetheless.

The last problem is a mechanical issue. Whereas other robots such as the TIAGo robot from PAL Robotics (12) (which is commonly used in the Open Platform League) has a robotic arm with seven degrees of freedom allowing it to reach almost every possible position easily. Our robot Pepper has only five degrees of freedom (Shoulder Pitch, Shoulder Roll, Elbow Yaw, Elbow Roll, Wrist Yaw) which means that when we give the target 6D point for grasping, there is a huge chance that pepper won't be able to reach it given its poor movement freedom.

2.2 Implemented Solution

The final implemented solution was an action server used to set the robot in various predetermined poses. In ROS, an Action Server is like a supervisor for robots. It helps them handle complex tasks that involve multiple steps and take more time. When the robot wants to start a task, it sends a message to the Action Server to begin the task. The Action Server listens and executes the order. While the robot is doing its task, it can give the Action Server feedback on what it is doing. The Action Server keeps track of this progress. Once the task is finished, the Action Server send a message to inform the the task is completed. If the robot gets stuck or needs to stop the task, it can tell the Action Server to pause or switch to something else. The Action Server understands and helps the robot adjust its plans.

The action server created for movement handled all things related to moving the robot's arm and body. Some poses have to be maintained for a long period of time, so the movement server creates threads to handle that. The possible actions are :

- Open or close the hand
- Raise, straighten or bend an arm
- Try to grasp an object with both arms
- Put the robot in a position where it is holding a tray (for the restaurant task), see Figure 5
- Put the robot in a position where it can receive and hold an item in his hands

Figure 5: Pepper holding a tray

3 Contribution : Other work

For the finals we wanted to present Louis Li's work. He recreated a virtual twin of Pepper in a VR environment. We wanted to showcase that to the jury. Our goal was to show Pepper's virtual twin interacting with objects in the competition arena. I was tasked with recreating the RoboCup@Home arena in VR. We used the unity game engine for the virtual reality environment. I recreated the entire layout of the arena, then using a 3d modeling tool called Blender, I created exact 3D replicas of all the furniture present in the arena.

4 Result

Thanks to all the team's work, we managed to win the RoboCup@Home Social Standard Platform League for the second year consecutively. We scored a total of 800 points in stage 1 and 200 points in stage 2. All the results are displayed in Figure 6.

First we scored 41.67 points for our poster. This is the best score for a poster in all the SSPL and in all the @Home league. Maelic did the poster presentation and did an amazing job at it.

For stage 1 we only scored in the receptionist task. This was the most crucial task, it's a task where pepper can use its social aspects the best. We managed to get 800 points for this task alone. We did not score in the storing groceries, carry my luggage, serve breakfast and GPSR tasks. Other teams did not score either, only the Colombian team, Sinfonia, managed to score 200 points in receptionist.

In stage 2 we only scored in the stickler for the rules task, which gave us 200 more points. In the clean the table task, Pepper was able to grasp a spoon with the help of an operator as seen on figure 7. Getting this far in the task is really rare, the public was impressed and gave us a round of applause. Sinfonia scored 400 points in stickler for the rules, so we still had a solid lead after stage 2.

After stage 2, our team and Sinfonia were the 2 best teams, so we could compete in the finals. However Sinfonia gave up and forfeited, not even taking part in the final although they were qualified, which is really unsportsmanlike.

4.1 Problems encountered during the competition

During the competition several problems occurred. First of all the network was really overloaded, so it was too slow for us to work properly with the robot. We were forced to plug pepper in and follow it with a very long ethernet cable.

A lot of what we planned had to be rewritten as we found some issue with plans and functions. There were some problems with navigation, making it so the robot would not move at the beginning of some tasks, resulting in a timeout and 0 points.

SSPL

	Team name (ranking)	Stage 1 + 2 points
1	RoboBreizh	2,085.33
2	Sinfonia Uniandes	1,266.33
3	LiU@HomeWreckers	61
4	SKUBA	54.00

Figure 6: Scores for stage 1 and 2 of RoboCup@Home : Social Standard Platform League

Figure 7: Our Pepper with a spoon in its hands during the Clean the Table task

Conclusion

The main goal was to make Pepper grasp an object without human help, however I was not able to achieve that. As stated before there are many issues that stop us from making Pepper grasp object, mainly mechanical problems that can't be resolved unless you buy a different robot. The base idea of using a point cloud segmentation algorithm, an algorithm to determine the grasp pose and MoveIt to calculate and execute the movement works on other robots that feature a true mechanical arm with a parallel gripper and a better depth sensor. There might have been alternate solution however I could not find any that would fit our robot. Another solution could have been to write a program similar to MoveIt but using Pepper's characteristics and with really high tolerance. However this requires a lot of skill in mechanical equations that I don't have.

The other goal to participate in Robocup is a success as we won. I had more of a supportive role in the team, doing small jobs and help whenever I could. Taking part in this competition was really a great opportunity and made me develop a lot of diverse skills like working under pressure. This internship made me gain a lot of experience in computer science and robotics. I had never worked in robotics before, so there was a lot of things to learn. Working in 2 different computer science laboratories in 2 different country made me discover the world of research in a unique way.

In the end, I'm really glad to have been part of team RoboBreizh. It has been an amazing experience and I'm really happy that all the work that the team put in made us win.

References

[1] C. Buche, M. Neau, T. Ung, L. Li, T. Jiang, M. Barange, and M. Bouabdelli, "RoboBreizh, RoboCup@Home SSPL Champion 2022," in *RoboCup 2022:: Robot World Cup XXV*, (Berlin, Heidelberg), pp. 203–214, Springer-Verlag, Mar. 2023.

[2] A. Dizet, C. L. Bono, A. Legeleux, M. neau, and C. Buche, "RoboCup@Home Education 2020 Best Performance: RoboBreizh, a modular approach," July 2021. arXiv:2107.02978 [cs].

[3] J. Hart, M. Matamoros, A. Moriarty, H. Okada, M. Leonetti, A. Mitrevski, K. Pasternak, and F. Pimentel, "Robocup@home 2022: Rules and regulations." `https://athome.robocup.org/rules/2022_rulebook.pdf`, 2022.

[4] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system." `https://www.ros.org`, 2018.

[5] D. Coleman, I. Sucan, S. Chitta, and N. Correll, "Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study," Apr. 2014. arXiv:1404.3785 [cs].

[6] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and Model set: Towards common benchmarks for manipulation research," in *2015 International Conference on Advanced Robotics (ICAR)*, pp. 510–517, July 2015.

[7] Y. Qin, R. Chen, H. Zhu, M. Song, J. Xu, and H. Su, "S4G: Amodal Single-view Single-Shot SE(3) Grasp Detection in Cluttered Scenes," in *Proceedings of the Conference on Robot Learning*, pp. 53–65, PMLR, May 2020. ISSN: 2640-3498.

[8] B. Zhao, H. Zhang, X. Lan, H. Wang, Z. Tian, and N. Zheng, "REGNet: REgion-based Grasp Network for End-to-end Grasp Detection in Point Clouds," Mar. 2021. arXiv:2002.12647 [cs].

[9] A. t. Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp Pose Detection in Point Clouds," June 2017. arXiv:1706.09911 [cs].

[10] M. Gualtieri, A. t. Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," June 2017. arXiv:1603.01564 [cs].

[11] S. Liu and P. Liu, "Benchmarking and optimization of robot motion planning with motion planning pipeline," *The International Journal of Advanced Manufacturing Technology*, vol. 118, pp. 949–961, Jan. 2022.

[12] "TIAGo - Mobile Manipulator Robot." `https://pal-robotics.com/robots/tiago/`, 2023.