THE UNIVERSITY OF QUEENSLAND

Use of Robotic Arm as a Tool in Manipulation Under Uncertainty for Dual Arm Systems

Student Name: Jimy Andre Cai Huang

Course Code: ENGG7290

Supervisor: Dr Surya Singh

Submission date: 10 June 2019

Faculty of Engineering, Architecture and Information Technology

Executive Summary

Despite the latest advances in research, one of the greatest challenges in robotics is making systems that are dextrous at manipulation tasks in real human environments. The main factor for the difficulty of this domain is having to deal with the uncertainties inherent in the nature of hardware components on a robot, and its surrounding environments.

In recent years, various methods have been proposed to tackle manipualtion tasks under the presence of uncertainty. Today, one of the popular methods is to frame manipulation tasks such as grasping or moving objects, under a sequencial decision making process known as the “Partially Observable Markov Decisision Process” framework, which allows a problem to be modelled in terms of a few components, and provides solutions in the form of policies that map a given problem state, to an optimal action the agent should perform to attain the most rewarding future outcome.

This report explores the possibility of using a robotic arm as a tool in a two-arm manipulator, to introduce benefits such as “uncertainty reduction”, and “object movement restriction” in manipulation tasks like grasping under the presence of uncertainty. In particular, the project discussed in this document seeks to develop models and methods to make a robotic manipulator able to perform concious use of one of its arms as a wall when an object is suspected to be at risk of falling outside of the designated working area, and as suporting structure to help the grasping arm push the target object into it and achieve a stronger grasp.

Theoretical analysis on the methods produced by this project, along with experimental results from testing on a real robot tasked to grasp a tube of pringles under the POMDP framework, indicate that a robot can be modelled to understand its situation and make a decision on whether it needs to use one of its arms to facilitate its grasping task, or if it is better off approaching the object directly.

Finally, further analysis of the methods produced in this project, present a good indication that such concious robot behaviour for using one of its arm as a blocking wall and facilitate the overall task of grasping, could be potentially generalized to grasping experiments of different objects, including objects that might present challenging for one-arm systems.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

The concept of autonomous robots simplifying our everyday living by taking care of tedious tasks such as household chores while we spend time on more satisfying activities, has been around since the early times of the robotics field. In order to conceive such systems, robots must achieve capabilities such as manipulating objects, and using tools to complete given tasks under real human environments. However, despite the latest advances in research, such fantasized systems are still not attainable. Robots today still struggle with manipulation tasks such as grasping or lifting objects under real environments where the operating scene is rarely structured and static. Without the dexterity for handling and interacting with objects, the usefulness of robots are still limited to industrial settings where the environments are usually tightly controlled.

The main problem of achieving robotic dexterity in manipulation is having to deal with noise arising from sources such as the nature of hardware components, modelling abstractions, and lack of knowledge on the environment surrounding the system. As a consequence, the execution of commands given to a robot and the sensory information it can obtain are not completely reliable, making known deterministic methods to planning a successful task solution obsolete. To overcome this issue, researchers have proposed different approaches for planning problems under the presence of uncertainty [1], [2], [3].

A popular framework for manipulation experiments under motion and sensing uncertainty, is the Partially Observable Markov's Decision Process (POMDP), a decision making construct that produces a mapping between a belief, which represents a picture of an agent's reasoning of its state at a given time step, to a best action that the agent can execute under that belief to maximise its expected total reward [3].Some manipulation studies under this framework can be seen on [4], [5], [6], and [7], with studied problems being more complex as algorithms for offline and online POMDP solving have improved throughout time [8],[9], [10], [11].

1.2 Problem Definition

Despite the advances on manipulation research, most studies that take into account the inherent uncertainty in real environments are still based on one arm systems. While literature is available for dual arm manipulation, most of existing studies [12], [13], [14], are still based on approaches that require deterministic behaviour or sensing from the robot.

1.2.1 Aims and Goals

The purpose of this project is to investigate some of the benefits that a dual arm system can bring in manipulation tasks under uncertainty. In particular, this project will build on a previous research involving planning for autonomous grasping of a cylindrical cup under the POMDP framework [7], to examine how an additional arm (left arm) can be used a tool to manipulate more "difficult" objects that one arm systems would normally have trouble with.

The goal of this project can be framed as achieving the following desirable behaviour.

Desirable behaviour: The dual arm robot being able to "see" the benefits of using its left arm as a wall (tool) in its decision making process and make use of it to increase its ability to manipulate more "difficult" objects. These benefits include:

- **Blocking:**
  The use of putting down the left arm as a wall can block the movement of the target object past a certain threshold such as the end of a table. This will prevent situations where an object manipulation task is no longer achievable because the target object is difficult to reach.

- **Uncertainty reduction:**
  By blocking the movement of the target object beyond the location of the left arm acting as a wall, also allows for uncertainty reduction. This occurs when the uncertainty surrounding the object's location includes positions past the wall. By the physical presence of the wall, the object cannot move past it when present. As a result, these areas of uncertainty can be eliminated, increasing the likelihood of the object's positions in all other regions.

An important remark to make for this desirable behaviour is that the robot should be able to weight in the time it requires to use the left arm as a wall against the need to use it. This is, if the target object is still far from going out of bounds and the robot can safely go for a grasp, then using the left arm will only delay the time required for the task, and the dominant behaviour in such scenarios should be to directly approach the target object and grasp it. Moreover, the nature of the target object must also be considered. For objects that are more likely to go out of reach (i.e slippery objects), the robot should consider the need to use the left hand sooner than for more static objects.

Figure 1 depicts an example of how an additional hand could be used as a barrier to block the movement of target objects past certain locations.

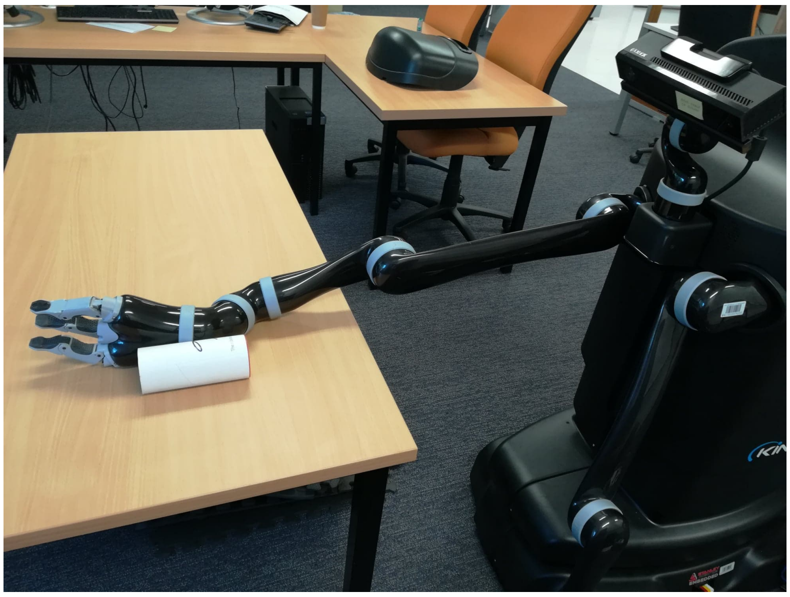


Figure 1: Use of robotic arm as a barrier

1.2.2 Scope

Due to time constraints, the primary focus of the project will be on achieving the desirable behaviour previously described. Therefore, initial experimentations will only consider manipulation of a constant category of objects. Residual time, if any, will then be used to expand experimentations across further categories of objects.
The scope of this scope extends to the following tasks:

- Modelling manipulation problem under the POMDP framework
- Implementation of the POMDP model in software
- Simulation of experiments in software
- Experimentation on physical manipulator system

The following objectives and deliverables further specify what is expected to be achieved in this project.

1.2.3 Objectives

All stages mentioned in the scope require domain knowledge on robotics and motion planning. Therefore, objectives that will allow a successful achievement of the project's aim are:

- Improve understanding on robotic planning under the presence of uncertainty.
- Acquire skills for implementation of robotic models and simulations under popular robotic software frameworks such as "Robotic Operating System (ROS)", "Gazebo Simulation Engine", "MoveIt Framework".
- Experimentation of autonomous manipulation task on physical system (MOVO).

1.2.4 Deliverables

Final Product: Upon successful completion of the project, the final product of this project is expected to have the following components:

- **POMDP Model:**
  The final POMDP abstraction of the manipulation task proposed.
- **Software implementation:**
  The software implementation for the experimentation of the manipulation task.

In the contrary case where the aim is not attained by the end of the project timeline, the deliverables listed above are still expected up to the functionality achieved. However, discussion of the things that went wrong and analysis of how things can be improved are required on a final report.

Moreover, the following additional deliverables with their corresponding deadline are expected to support documentation of progress and milestones to the final product.

- **Monthly reflections (8th of February 2019, 13:00 - 8th of June 2019, 13:00:**
  Self reflections on the performance fo the work done so far and improvements as needed.
- **Initial Proposal (7th of March, 2019 13:00):**
  Document describing the details of the project, a literature review on similar work, and a project plan to provide a timely control of the work to be done.
- **Interim Report (2nd of May, 2019 13:00):**
  Document describing progress on the work to date and any changes to the initially proposed project.
- **Final Report (10th of June, 2019):**
  A final document detailing the results and learning obtained from the project along with recomendations for future work.
- **Oral Presentation (13th of June, 2019 15:00):**
  A presentation of the results and learning from the project to date.

1.2.5 Project Plan

A summary of the project timeline work structure, relevant risk analysis and opportunities for this project can be found on appendix “A” of this report.

2 Technical Background

The following presents a brief survey of some popular frameworks:

2.1 Markov's Decision Processes (MDP)

When the uncertainty of the task problem is only on the actions of the robot, and its state information can be fully derived or sensed without errors, a common framework to plan for the agent's behaviour is known as the Markov's Decision Process [15]. This framework abstracts a problem into components $< S, A, T, R >$ where:

- S (State Space) Describes the set of all possible states the robot can be in at any particular time step. A state describes relevant information that needs to be maintain at each decision making step of the robot. This could vary from the poses or joint angle information of the robot's components to encoded descriptions of objects in the environment.

- A (Action Space): Describes the set of all possible actions that the robot can execute. Elements of this set are user defined such as moving a robot arm to particular configurations.

- $T(s, a, s')(TransitionModel)$: Describes a model of the environment's world dynamics, represented as a probability distribution on ending up at a state s' given an action "a" was executed from a state "s"

- $R(s, a)$: Describes a numerical reward an agent receives by taking an action "a" from a current state "s". This component guides the behaviour on the robot by dictating what is good and bad behaviour.

The solution in this framework is given as an policy $\pi^*(s) \to a$, a mapping between states and actions, which dictates the best action that the agent can take at a given state or situation, in order to achieve the most rewarding expected outcome.

At any given state, the optimal value function is iteratively given by

$$V^*_\pi(s) = max_a[R(s, \pi(s)) + \gamma \sum_{s' \epsilon S} T(s, a, s')V^*(s')] \tag{1}$$

And with an optimal value function, an optimal policy is given by

$$\pi^*(s) = argmax_a[R(s, \pi(s)) + \gamma \sum_{s' \epsilon S} T(s, a, s')V(s')] \tag{2}$$

2.2 Partially Observable Markov's Decision Process (POMDP)

More realistic scenarios under which robots operate are commonly not fully observable. Instead, uncertainty is present in the nature of the sensors, which limits a robot's ability to completely determine the information on its current state. To operate under such conditions, researchers often plan under an extended version of the (MDP) framework known as the Partially Observable MDP (POMDP). As an extension of MDP, POMDP abtracts a problem as components $< S, A, T, R, O, Z, \gamma, b_0 >$ where the first four components are the same as an MDP and the additional components manage the uncertainty in sensing by the following:

- O (Observation Space): Describes the set of all possible observations that the robot can obtain from its sensing. Similar to the state space, an observation represents relevant information that the robot can derive from its sensors. This includes poses obtained from a visual system, or joint values obtained from the robot's arm encoders.

- $Z(s', a, o)$: Describes an observation function as a probabilistic distribution on seeing an observation "o" conditioned by an action "a" and a resulting state " s' ".

- γ describes a discount factor to ensure convergence of the value function from a belief

- b_0: An initial belief denoting a probabilistic distribution on the possible starting state of the agent.

Since the state information of a robot cannot be fully determined in this extended framework, the situation the agent is in at any time step is represented a probabilistic distribution over the state space, where the value assigned to each state on the distribution represents the certainty of the agent of being at that particular state. This distribution is known as a belief, and the space of all possible distributions or beliefs, is known as the belief space. With this new extension, the framework can be treated as an MDP model where the state space becomes the belief space and the a solution for problems under this framework is given as a mapping $\pi^*(b) \rightarrow a$ from the robot's current belief state to a best action that the robot should perform to maximize the expected reward it can attain [3].

2.3 POMDP Solvers

Although the POMDP framework manages to capture uncertainty for tasks planning, their solutions are often computationally unfeasible for medium to complex sized problems. This unfeasibility stems from the extensive computation required to explore the different paths that the robot can take from a given state, also known as the "curse of dimensionality", and the depth level of the paths branched, known as the "curse of history". Over the years, significant improvements have been seen on POMDP solving algorithms as attempts to overcome the aforementioned "curses" by using techniques such as approximating the belief space by a smaller set of representative points [8] [16] and smart sampling strategies that focus on the most promising areas of the belief space [9] [10] [11].

Despite the possibility of computing POMDP solutions offline (before run time of the robot) when experimenting with autonomous manipulation, which allows for longer computational times to approximate an optimal policy, doing so often requires solutions to models of each of the different environments that the robot might encounter. These models are often unknown a-priori due to the presence of uncertainty. Instead, in practice only models the known parts of the starting environment and relies on online solvers to allow the robot to adapt to changes in the environment. Recent work on general online solvers that can handle dynamic environments [10] [11] showed promising enhancements to this approach by introducing a method to recycle valuable computed information upon environment changes compared to previous approaches which recomputed solutions from scratch for the new environment model.

2.4 Software frameworks for POMDP Implementation

With advances in the algorithms and solutions for planning under the POMDP framework, software implementations of these are necessary for appropriate experimentation. Recent software frameworks that implement promising solvers [10] [11] are offered by the same research group that proposed them. The newest of the two, OPPT (Online POMDP Planning Toolkit) [17], proposes a modular framework that attempts to reduce software implementation time of their experiments, allowing for greater focus on the methodologies and algorithms. To achieve this, the toolkit was designed using a plugin architecture similar to popular robotic frameworks to reduce the initial learning curve, and allows POPMDP component definitions via a brief configuration file.

Moreover, the toolkit allows flexible configurations or customizations on the solving method used, and provides interfaces to ROS and Gazebo high fidelity simulations and interactions with physical systems.

3 Literature Review

3.1 POMDP Manipulation

Research over the recent years have looked at targeting manipulation tasks under uncertainty by framing these tasks as sequential decision making processes under partial observatibility, this is, under a POMDP framework. One of the inital research for in this area was [4], which adopted use of "Heuristic search value iteration" as the POMDP solver to generate policies for two problems, placing a finger on a stepped block, and using two fingers to grasp a block. Despite the impressive results shown in this work, this approach could only cater for relatively small problems. Moreoever, the results presented were based on exprimentation by simulation rather than real life execution. Migration form the simulated setting to the real life setting can affect the results as the inherent uncertainty present is often higher.

Another study, [5] addresses grasping objects by using tactile sensing as a mechanism to reduce the uncertainty about the location of the object and reducing the problem complexity by limited the size of the action space. This limitation is achieved by computing movement behaviours for the end-effector as "macro actions" in an offline fashion, instead of framing the action space as primitive movent of the robot's joint angles, and then selecting among the different pre-computed behaviours in real time execution. The study experimented their methods in both a simulated environment and on a real robot, showing the promising results can be achieved even for shallow lookead depth on the search at planning time.

Finally, one of the more recent studies [7], explores the possibility of using planning under the POMDP framework as an application for a robotic candy serving agent. To do so, the different manipualtion tasks involved in a robot serving candy such as grasping the candy container (a cup), moving the container to the candy area, scooping the candy with the cup, and bringing the cup full of candy to a designated location, were framed as individual POMDP problems. Solutions for these POMDP problems were generate in an online manner during real time execution to generate policies that dictate the best action for the robot at each time step.

The studies described above along with other similar studies undertaken over the last few years have demonstrated that manipulation tasks under the presence of uncertainty can indeed be achieved by framing these tasks as problems under the POMDP framework.

3.2 Bimanual-manipulation

Another important domain related to the purposes of this project is the use of dual arm systems in manipulation tasks. Particularly, the area of “bimanual-manipulation”, which focuses in manipulation tasks where both arms interact with the target object to achieve the goal.
Research on the literature shows a variety of approaches to bi-manipulation.

3.2.1 Deterministic motion planning methods

One of the approaches seen is the extension to deterministic motion planning methods. For instance, [12], investigated the handling of common household objects that are normally manipulated facing upward such as food and oven trays. This upward facing constraint information was exploited to derive a clever heuristic, allowing the planning problem to be addressed using a deterministic tree search planning method.

Another study, [13] discusses an experiment using extensions to known sampling based planning methods and inverse kinematic calculations to compute suitable trajectories for achieving grasping and re grasping on their dual arm robot.

Similarly, [14] presents a pipeline for the visual detection and grasp planning of objects on dual arm systems by enhancing the popular STOMP method, stochastic trajectory optimization for motion planning, to generate feasible grasping plans.

Despite the reasonable results seen in [12], and the ability of [13] and [14] to generate collision free motions for their dual-arm systems, the algorithms are restricted by limitations such as having to know the target’s pose beforehand or relying on accurate behaviour of the robots.

3.2.2 Master-Slave systems

Other approaches to bi-manipulation include a master-slave system where a slave robotic arm follows the master arm at a relative constrained space that conforms with collision free paths or closed-kinematic chains. One work under this category, [18], examined a visual guided system where the slave arm tracks the master and maintains a constant relative pose to tackle grasping and moving of objects. Another work, [19] adopted the master-slave system to explore force control and shared force strategies when manipulating objects. Despite the popularity on the master-slave system, the approach requires deterministic information on the state of the arms to compute feasible paths and tracking of the slave arm. This constraint creates a gap in the ability to cater for uncertainties of motion.

Although more approaches exist in the current literature for robotic bi-manipulation, most of the studies stay in the deterministic planning domain. These methods do not tend to scale well when the tasks are executed outside of controlled settings due to the inherent uncertainties of human environments and hardware from which the robots are built from.

3.2.3 Studies with uncertainty

One recent study that addresses uncertainty, shows a method using a two-armed system to reach for moving objects while accounting for the uncertainty in the motion of the objects [20]. The method proposes the use of a virtual object to couple task and coordination constraints into a close kinematic chain. This allows for planning for simultaneous reaching of the arms to predetermined reaching points in the object of interest. A problem with the approach presented in this study is that the state of both arms are assumed to be known at each time step, ignoring uncertainty that might arise from the motion of the robotic arms as it approaches the moving object.

3.3 Robots using tools

A final domain worth exploring for the purpose of this project is on robots using tools to interact with their environment. Different approaches have ben employed throughout the years to investigate the capabilities of robots using tools.

One study, [21], invesitgated how multiple robots can behave cooperative to use sticks and strings to align and group objects on their surroundings. Although results indicated that robots can be used cooperatively to interact with multiple objects in the environment at a time, the motion planning strategies adopted in this study required a structured and controlled environment, where the location of the different objects was assumed to be known. This makes the methods investigated not very scalable to unstructure environments, where robots must be able to see the value of using tools based on their current circumstances.

More general studies, like [22], investigated methods to learn the different outscomes of interacting with a tool in the environment and how these learned interactions can then be used to make a robot use tools in order to accomplish tasks.

Further studies in the more recent years have attempted teaching robots how to use tool from human demonstration. The use of imitation learning helps on overcoming the strict methods that require static knowledge of the models in the environment, and systematic approximation of real world behaviours in software implementations. One study in this area [23], investigated how to generate humanoid body motions corresponding to the use of different tools, such as a wooden sword and a tennis racket, by demonstrations under partial observability.

As a further extension and more general method to the aformentioned studies, [24], explored methods to help robots improvise the use of unknown tools, this is, tools that the robots have not seen before and therefore, have no knowledge about. This study explores such challenge by use of machine learning to develop models that encode both visual and physical understandings of the behaviour occuring on interactions between objects in the environment.

4 Project Methodology

4.1 Project Resources

As this project sought to extend the functionalities of the study in [7], the design and implementations adopted for this research were based on the same software frameworks. Moreover, physical experiments were carried using a similar physical manipulator. All the required resources for the proper development of the project were available and easily accessible.

4.1.1 Physical system for mobile manipulation

The physical manipulator used for this project was a mobile manipulator from "Kinova Robotics".

For the manipulation tasks investigated in this project, only the two "7-DOF" robotic arms and the "Kinect" camera were used. More technical details of the system can be found on `https://www.kinovarobotics.com/en/products/mobile-manipulators`. Moreover, although the system was equipped with robotic arms with seven degrees of freedom, in this project, one degree of freedom was kept fixed. The reduction of a seventh joint on each arm was done to lower the size and complexity of the POMDP problem.

4.1.2 Software frameworks and packages

- Robotic Operating System and GAZEBO Simulation Engine: These are popular software frameworks for implementing and simulation robotic implementations. Access to these frameworks are open to the general public.
- OPPT framework: This is freely distributed to the general public and mantained by the research group in which this project is to be developed.
- Kinova software packages: These are provided by Kinova Robotics and are freely distributed to the general public.

4.1.3 Learning resources

- Relevant learning material: Learning material such as conference publications or journal papers were accessible from the University's library partner databases.
- Mentoring and advise: Internal and external advisors are available for mentoring, problems and gaps of knowledge related to the project.

4.2 Technical Approach

The following summarizes the methodology adopted to ensure the successful completion of the project.

4.2.1 Develop a problem model

The first stage for addressing the problem was to abstract it into a relevant model. Since this project sought to extend the functionalities of the study in [7], the problem was also framed as a “Partially Observable Markov’s Decision Process (POMDP)”. Therefore, this first staged focused on representing the dual arm manipulation problem into a set of components $< S, A, T, R, O, Z, \gamma, b_0 >$. The final iteration of the model used is presented in the results of this report.

4.2.2 Software implementation of Problem in OPPT

Following the modelling phase of the problem, the next stage of work was on the software implementation of the POMDP problem using relevant software frameworks.

The general implementation of the problem was made using a POMDP software toolkit known as ”OPPT” [17]. Using OPPT reduced the software implementation of the problem down to specification of POMDP components through isolated plugins from the core OPPT toolkit. Once the problem was defined, the toolkit included a default integrated solver [10], to generate policies as approximate solutions to the decision making problem.To ease the simulation aspect of the software implementation, this project adopted the use of ”GAZEBO”, a popular robotics simulation engine which could be interfaced from within the OPPT software toolkit.

Model files required by both ”OPPT” and the ”GAZEBO” engine for a detailed representation of the robot were sourced from ”Kinova Robotic’s” official resources. Figure 2 shows an example representation of the physical manipulator used for experimentation in this project.

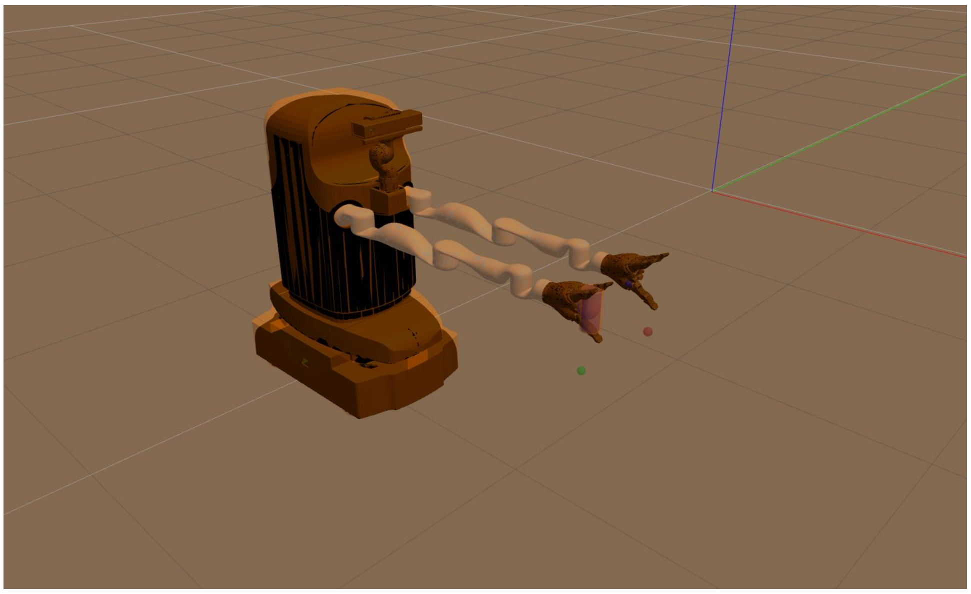


Figure 2: Collision model of physical manipulator in GAZEBO simulation environment

Although original model files allowed for the physical manipulator allowed for more accurate modelling in software and access to more detailed information, modifications were made to these files to facilitate implementation of the problem and overall improve performance in the motion planning. Figure 3 shows the modified version of the robot where the links of both arms were approximated by cylinders. This modification allowed for faster collision checking time for the different sections of the robots arms by decreasing the time complexity on checking for collision in complicated triangular meshes, to objects with simpler geometry.

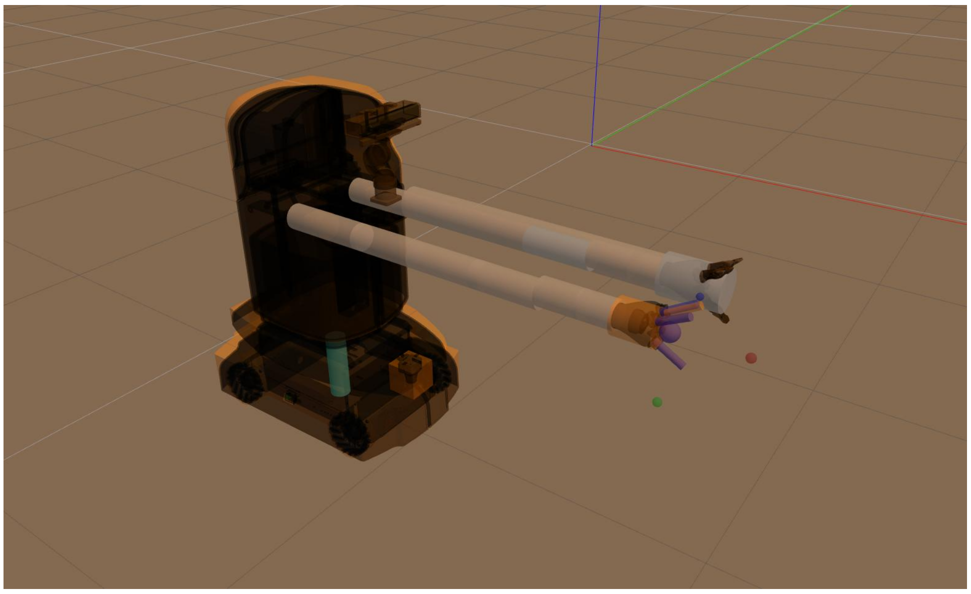


Figure 3: Cylindrical model of physical simulator in GAZEBO simulation environment

4.2.3 Simulation Testing and Analysis

For simulation runs of the experiments, the OPPT toolkit includes a simulation feature that interacts with the a GAZEBO interface and updates the state of a simulated environment upon an action execution of the robot. Relevant information at each step of the experiment was recorded in log files that could later be used for analysis and interpretation of the robot's reasoning at particular trial runs. To further facilitate the analysis on simulation runs, a small program was written to "play back" any experiment, and plot the uncertainty information seen by the robot at each step.

Figure 4 below shows an example of a play back step for a simulated experiment. The left part of the figure shows how the robot looks like at the selected state, while the right part of the figure shows the uncertainty with respect to the object's position in the correspodning current belief of the selected time step. The red points on the plot in this example show the position at each of the different object positions seen on the states making up the belief state of the robot, while the blue point is the simulated actual position of the object.

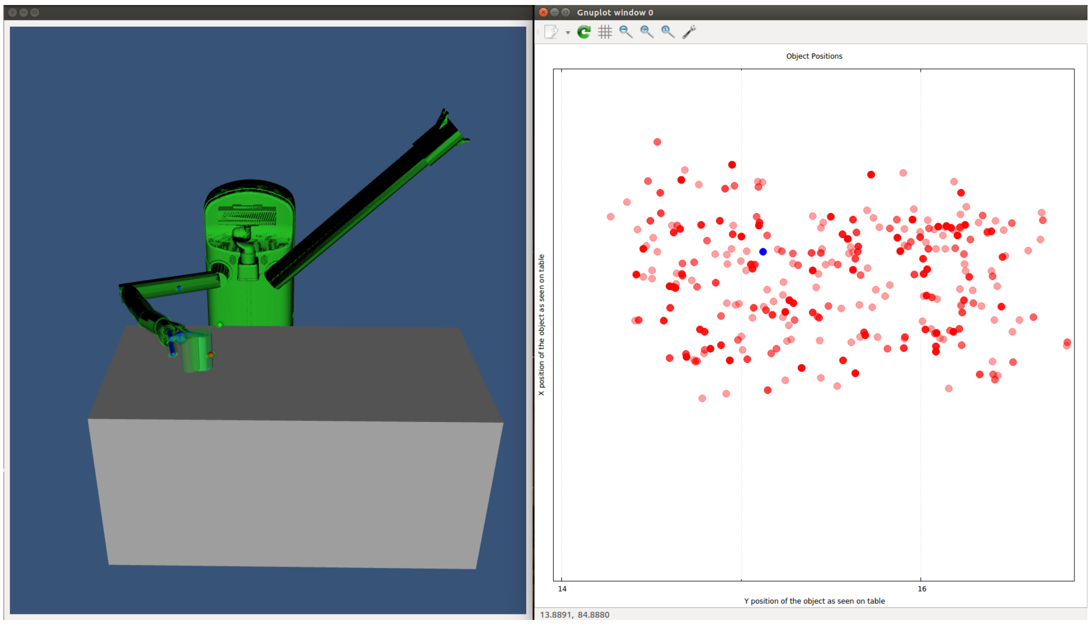

Gnuplot window 0
Object Positions
X position of the object as seen on table
14
16
Y position of the object as seen on table
13.8891, 84.8880

Figure 4: Simulation and uncertainty playback tool

4.2.4 Experimental testing with physical manipulator

Following the simulation phase, experiments were moved to the physical manipulator. The software-hardware interface adopted in [7] to communicate with the robot was expanded to include functionalities for the additional arm introduced in this study.

Experiments with the real robot were carried out inside the laboratory. A table was used as the experimental scene and the target object used for grasping was a tube of pringles. Figure 5 below shows the experimental scene under which the grasping trials were run.

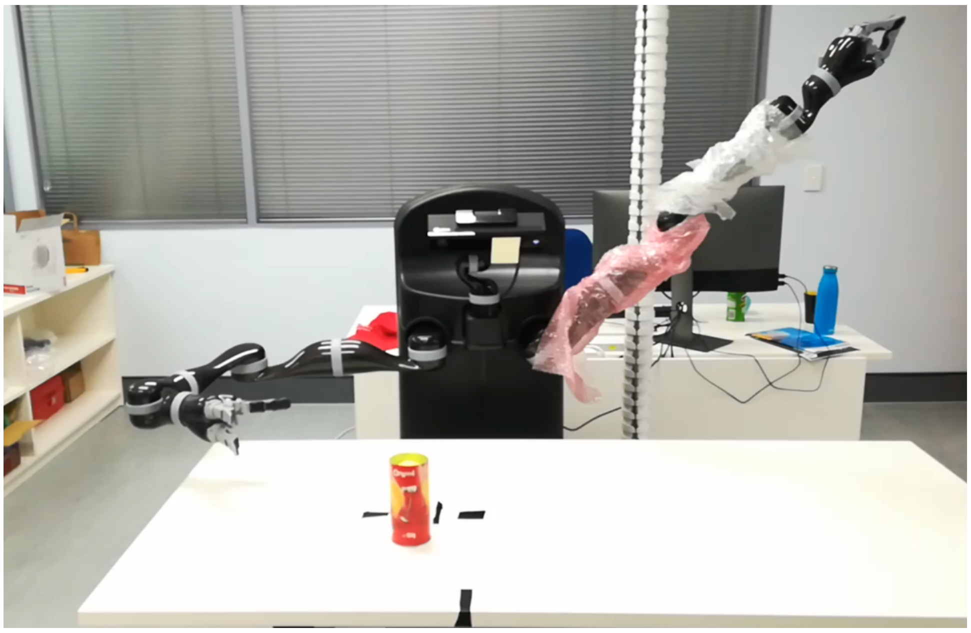


Figure 5: experimental setup for physical manipulator

To aid on the debugging and understanding of the robot's behaviour, plots similar to the one in figure 4 were generated live during any robot experiment trial. Finally, similar to the simulation testing, relevant information about the robot and its decision making process was logged at each trial run so that it can be reproduced at a later time.

4.2.5 Analysis and Revision

The last technical stage of this project was to analyse the results obtained and revise the algorithms and implementations as necessary to achieve the desired results.

5 Results and Analysis

The results and analysis of the work described in this project can be categorized into the following subsections:

- The final product discussed in the 1.2.4 of this report. This subsection discusses the model and methodologies produced thorugh experimental iterations and algorithm analysis to achieve successful behaviours in experimentation runs.
- Experimentation Results 5.2.1 This subsection discusses and analyses the quality of the experimentation results on the manipulation task obtained from by use of the intelectual property and software implementations included in the final product.

5.1 Final Product

At this last stage of the project, the deliverables in the final product proposed in 1.2.4 have been completed. The following describes and analyses the final versions of the deliverables and methodologies that integrate the final product.

5.1.1 Software Implementation and Simulation

All current software implementations relevant for experimentations of the manipulation task, including the different POMDP components as implemented plugins on the Online POMDP Planning Toolkit (OPPT) and the software interface for the control of the physical system can be found on the following version control repositories.

- Progress on the POMDP problem implementation for this project can be found on the "movo_2arm" branch of this repository `https://github.com/RoboticsDesignLab/oppt_devel/tree/movo_2arm`
- Progress on the software and physical system interface can be found on the "interface_2arm" branch of this repository `https://github.com/RoboticsDesignLab/popcorn`

5.1.2 POMDP Model

The following presents the current version of the problem model, framed as a POMDP. This model was the result of various iterations and experimentations with each new version covering important abstraction gaps from the last versions.

- **State Space "S":** The state space provides information on relevant information for the robot at any particular time step.

  The state space for this project was given as the Cartesian product of the following spaces:

  - Target Object's pose: A 6D space representing the pose of the coordinate frame attached to the target object relative to a world frame.
  - Right Arm Configuration(θ_{RA}): A 6D vector describing the angles of all the different joints on the right arm.
  - Left Arm Configuration (θ_{LA}): A 6D vector describing the angles of all the different joints on the left arm. The spaces of joint angles that achieve a"WALL" (left arm down) or a "HOME"(left arm up) configuration will be referred to as"S_{WALL}" and "S_{HOME}" respectively.
  - A binary space {LEFT WALL, LEFT HOME} indicating if the left arm has been set to either a "wall" or a "home" configuration.
  - A binary set {OPENED, CLOSED} indicating if the right hand gripper is opened or closed.
  - A binary set{GRASPED, NOT GRASPED} indicating if the robot has grasped the target or not.

  Although the size of this state space is considered to be relatively large, the POMDP solver adopted in this project ("ABT") is based on a sampled based methodology, making it capable of handling large state spaces by considering only promising states from a particular point.

- **Action Space "A":** The action space describes the set of all the possible actions available for the robot to execute.
  The action space for this project was chosen as the union of the following action sets:

  - **Right Arm Action Set (A_{RA}):** A six dimensional set describing a constant value increment/decrement for each of the 6 joint angle of the right arm considered in the state space.
  - **Gripper Action Set (GA):** A binary set {OPEN, CLOSE} where
    - OPEN: Sets the finger joints of the right hand to an OPEN configuration.
    - CLOSE: Sets the finger joints of the right hand to a CLOSE configuration.
  - **Left Arm Actions (A_{LA}):** A binary set {WALL-UP, WALL-DOWN} where

    * WALL-UP: Describes the action of lifting the lower arm back to a HOME position. Initially, this will be a predetermined fixed pose away from the operating scene. When this action is performed, the joint angles of the left arm are set to a "θ_{HOME}" configuration.
    * WALL-DOWN: Describes the action of positioning the lower arm to a WALL configuration on the operating scene. When this action is performed, the joint angles of the left arm are set to a "θ_{WALL}" configuration.

  The size of the action space is therefore given as $|A| = 2^6 + 4 = 68$ where the initial term of the sum is equal to the possible different combinations for constant increments and decrements of all six joints and the second term account for the extra individual actions. Although a large action space increases the complexity of the POMDP problem significantly, the individual joint increment and decrements allow for greater choice and freedom of movement of the robot to accomplish the task.

- **Transition Model T(s,a,s'):** The transition model describes the world dynamics of the environment. In other words, this component models how a particular state " s " is altered into a next state " s' ", when the robot executes an action " a ". The transition model adopted for this project is described by the following:

  Executing a right arm action: Choosing a "Right Arm Action" results in a linear incremental of the right arm joint angles according to $\theta' = \theta + a_\theta + e_\theta$, where θ' and θ are the next and current joint angles of the right arm respectively, a_θ is a 6D vector of joint angle increments according to the action applied, and e_θ is an additive control error.

  A right arm action results in movement of the right end effector. If a collision occurs between the right arm and the target object, along the movement trajectory, the target object moves along with the colliding section of the right arm.

  Executing a gripper action:

  - Choosing an open gripper action results in a deterministic opening of the gripper. This is, any next state " s' " resulting from an "open gripper" action will have its "Gripper Status" information set as "OPENED".
  - Choosing a close gripper action closes the finger and sets the "Gripper Status" of its next state as "CLOSED". Moreover, closing the gripper also leads to the following conditional behaviours:
    * For states " s " where the object is within the necessary conditions for the grasp, (i.e, the end effector is sufficiently close to the object), the next state " s' " will set its "Grasping Status" as "GRASPED" with a specified probability p.
    * For states " s " where the object is not within the necessary conditions for a grasp to occur, the next state " s' " will set its "Grasping Status" as "NOT GRASPED".

  * For all other states where the object is within the necessary conditions for a grasp, but the grasping fails, the next state ' s' " will have its object location changed. This behaviour was incorporated to cater for scenarios where the object might be pushed by the robot when trying to perform a grasp.

Executing a left arm action: Taking a Left Arm Motion action yields a state where the joint angles of the left arm are set accordingly to configurations in the S_{HOME} or S_{WALL} spaces.

- Case a == A_{LA}.GO-HOME:
  $s'.Q_{LS} = HOME \qquad s'.\theta_{LA} = \theta_{HOME}$
- Case a == A_{LA}.SET-WALL:
  $s'.Q_{LS} = DOWN \qquad s'.\theta_{LA} = \theta_{WALL}$

Transitions by taking a "non left arm" action from a state where "θ_{LA}" ϵ S_{WALL} (wall is DOWN) must yield a state s' where the object is on the same side of the wall as the arm. This is because the object will be blocked past the "wall" if it attempts to displace past it. Moreover, actions of such nature do not affect the configuration of the left arm. More formally, this is given by:

Let the position of the left arm be at (x,y,0) relative to "f_{world}(world frame)" when "θ_{LA}" ϵ S_{WALL} then, a non-left arm action (case DEFAULT) must yield an s' with object position (x',y',0) relative to "f_{world}" where $y' < y$.

This behaviour shows the scenarios where the "uncertainty reduction" effect from the use of the left arm as a wall can be appreciated.

- **Observation Space "O"** The observation space is given as a subset of state space and is form by the Cartesian product of the following spaces:

  - Target Object's pose: A 2D space representing the 2D location of the coordinate frame attached to the target object relative to a world frame.
  - Right Arm Configuration(θ_{RA}): A 6D vector describing the angles of all the different joints on the right arm.
  - A binary space {LEFT WALL, LEFT HOME} indicating if the left arm has been set to either a "wall" or a "home" configuration.
  - A binary set {OPENED, CLOSED} indicating if the right hand gripper is opened or closed.
  - A binary set{GRASP, NO GRASP} indicating if the robot has grasped the target or not.

Contrary to the state space, the size of the observation space does significantly impact the performance of the POMDP solver. As a measure to reduce the complexity inccured from this observation space configuration, the space was discretized by a constant value relative to the information regarding the 2D position of the object's pose. In other words, two observations were considered the same if the relative distance between the 2D position of the objects on each observation was smaller than a certain threshold.

- **Observation Function Z(s',a,o):** For the observation function, the following is assumed:
  - The gripper encoder sensor is perfect. This is, after any grasping action, the robot can know with certainty whether the target object was grasped or not.
  - The joint angle encoders for the right arm are assumed to be disturbed by an additive error drawn from a uniform distribution with support $[-0.05rad, 0.05rad]$.
  - The observation on whether the left arm has been set as a wall or not, is perfect.
  - Observations on the 2D location are currently obtained at every time step. However, these observations are noisy are assumed to have additive error drawn from a uniform distribution with support $[-3cm, 3cm]$.
- **Reward Function:** The reward function guides a robots decision making by indicating which scenarios are more rewarding and which are worse to be in. The reward function for this project was adopted as the following:
  - The agent receives a reward of 2000 for a successful grasp. A high reward incentivizes the robot to reach out for the goal.
  - The agent receives a penalty of -250 for every collision with the environment. This penalty excludes the specific event where a robot collision happens between the inside of the right gripper and the object.
  - A penalty of -700 for the case where the gripper is closed without a grasp, and the robot does not immediately executes the openGripper action.
  - A penalty of -250 is the gripper is opened when a grasp is established.
  - A penalty of -200 for states where the object is "behind" the gripper.
  - A penalty of -200 when the object is at a position beyond a specified threshold, where the positioning of the left arm wall is useful.
  - A penalty of -100 when the roll and pitch rotation values of the end effector exceed a certain threshold. This is to encourage the end effector to approach the object with the best orientation.
- **Discount Factor:** $\gamma = 0.95$
- **Initial Belief:**
  The initial belief consists of a initial state with added error. This additional error is sampled from an uniform distribution and distorts the object's pose, and right arm joint angles of the initial state.

5.1.3 Rollout strategy

Although not explicitly a component of the POMDP framework, popular online solvers today make use of a rollout strategy to "$Q(b,a)$ " value associated with a belief state, when the corresponding action "a" is first explored from that state. These values encode information regarding the future expected discounted rewards that can be achieved by taking an action "a" from a belief state "b" and assuming that the robot behaves optimally in terms of its future rewards.

Rollout strategies become of paramount importance for popular online solvers in practice as the size of a problem grows in terms of its action space and observation space. Due to the limited planning time often available for planning in an online fashion, large problems often lead to shallow explorations during planning, which generally results in bad estimates of future rewards without the use of a rollout strategy.
To help guide the planning to more promising parts of the search, this project adopted the following rollout strategy:

$$\hat{Q}(b,a) = [\gamma^{t_{exp}} P_{ES} R_S] - [P_{EF} R_F + t_{exp} * stepCost] - [d_{pos}(o,ee) * stepCost] \quad (3)$$

- γ: The discount factor value used in the POMD model
- t_{exp}: The expected amount of grasping attempts until a successful grasping attempt is seen
- P_{ES}: The probability of eventually reaching a successful grasp from a particular object location
- R_S: The immediate reward for a successful grasping defined in the reward function of the POMDP model
- P_{EF}: The probability of eventually going out of bounds for the allocated operating area of the experiment
- stepCost: The magnitude of a step penalty defined in the reward function of the POMDP model
- $d_{pos}(o,ee)$: The euclidean distance between the object's position and the position of the end-effector

The expression in equation (3) assumes that the optimal future behaviour for the robot from a current state will be to approach the object and then attempt to grasp it. Since a grasp can fail according to the POMDP model described in the previous subsection, and the object could be displaced in such case, the optimal future behaviour of the robot is assumed to repeat the approach-grasp cycle until either the goal is reached or the objects is displaced out of bounds.

The first component approximates the positive reward seen in the future by considering how probable is it for the robot to reach the goal from the current state and estimating how long it will take the robot to do so. The other two components of this rollout attempts to account for the penalties that the robot might experience in the future from the current state assuming it behaves optimally. These penalties include the steps cost received for the robot to approach the object and perform the grasp action in an approach-grasp cycle, and the cost associated with scenarios where the object goes out of bounds.

To calculate the values of P_{ES}, P_{EF}, and t_{exp}, the grasping attempts by robot from a particular state were framed as a Markov Chain where each state of the Markov chain indicate the position of the object in a particular grid of discretized operating region, with the right end effector of the robot begin within necessary conditions (close enough) to have a chance of completing a grasp. Additionally, two more states "SUCESS" and "FAIL" are added to the chain for representing events where a grasp is completed, or the object falls out of bounds.

The transition model of the Markov chain was modelled as the following: At each state on the grid the right end effector of the robot is assumed to be within grasping conditions and the action perform is a grasp.

- For states where the left arm is not used (UP):
  From any state, the probability of successfully grasping, and therefore, transitioning to the "SUCCESS" state is given by a value p, and the probability of not completing a grasp is given by probability $(1 - p)$. Upon an unsuccessful grasp, a state transition can occur to the immediate state at the NORTH EAST direction, the immediate state at the EAST direction, and the immediate state at the SOUTH EAST direction with uniform probability. In this example the probability of transition would be given as $(1-p)/3$. If one of the states that can be transitioned into is "OUT OF BOUNDS" denoted by the red region in figure 6, this probability is added to the probability of transitioning into a the "FAIL" state instead.

  The following shows an example of a markov chain where the operating region is discretized into a 3x3 region but the left arm is not in use. The view of this example is assuming the observer views the operating region by facing to the front of the robot, this is, the observer and the robot face each other.

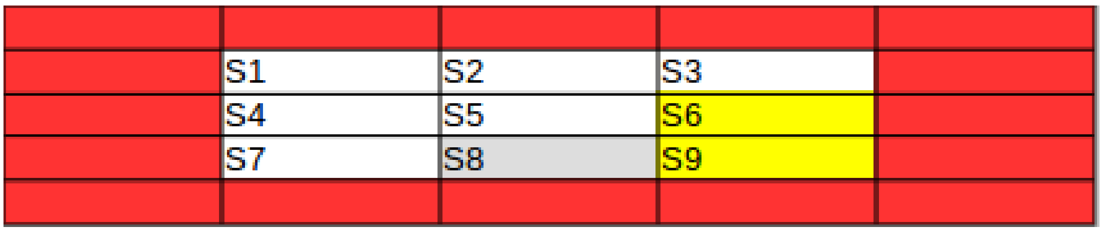


Figure 6: Example of markov chain transition model with left arm (UP)

- For states where the left arm is used (DOWN):
  From any state not immediately followed by GREEN region immediately to the right, as shown by figure 7,the probabilities of state transition are the same as the behaviour for "states where the left arm is not used (UP)". On the other hand, for states that are immediately followed by GREEN regions to the right, the probability of successfully grasping is modelled by a constant "$p = 0.95$", and the probability of not completing a grasp is given by " $p = 0.05$". This is to model the scenarios where the left arm serves as a supporting structure for the right arm of the robot to push the object against, facilitating the changes of grasping.
  Moreover, upon an unsuccessful grasp in these special states, a state transition can occur to the immediate state at the NORTH direction, the immediate state at the SOUTH direction, and to the same current state with uniform probability. These transitions result from the left arm blocking the object's movement, making a transition in the EAST direction not possible.

  The following shows an example of a markov chain where the operating region is discretized into a 3x3 region and the left arm is in use.

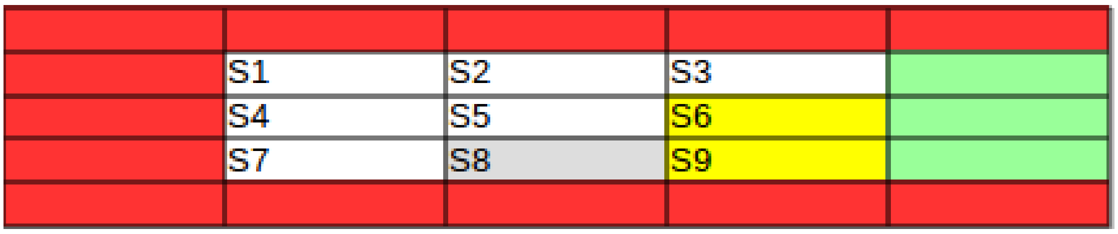


Figure 7: Example of markov chain transition model with left arm (DOWN)

With the Markov chain models of the grasping attempts for the robot shown above, the values of P_{ES}, P_{EF}, and t_{exp}, are given by the absorption probability to a "SUCCESS" state, a "FAIL" state, and the expected time to absorption to a "SUCCESS" state correspondingly.

Induced heuristic mapping from rollout strategy

The following tables show the discretized mapping values generated for a 5x5 discretization of a 40cm x 40 cm working area. These tables were generated by setting the parameter in the markov model to be "$p = 0.8$", and using the corresponding reward values for the "SUCCESS" and "FAIL" states from the reward function in the POMDP model. The values represented in each table correspond to the sum of the first two components of the rollout function presented in equation(3) when the left arm is not set down as a wall for blocking, and when the left am is set as a wall respectively.

1705.19	1703.97	1697.17	1658.25	1422.50
1806.82	1804.81	1794.00	1735.98	1422.50
1812.10	1809.82	1797.86	1735.98	1422.50
1806.82	1804.81	1794.00	1735.98	1422.50
1705.19	1703.97	1697.17	1658.25	1422.50

Table 1: Left arm is not set as a wall (UP) for p = 0.8

1705.52	1705.77	1707.42	1719.14	1822.59
1807.36	1807.71	1809.56	1819.22	1849.65
1812.71	1813.05	1814.61	1820.57	1849.99
1807.36	1807.71	1809.56	1819.22	1849.65
1705.52	1705.77	1707.42	1719.14	1822.59

Table 2: Left arm is set as a wall (DOWN) for p = 0.8

These values are interpreted from the an observer's perspective facing the robot. The following figure shows an example of the working area discretization and how the values in the table are matched accordingly.

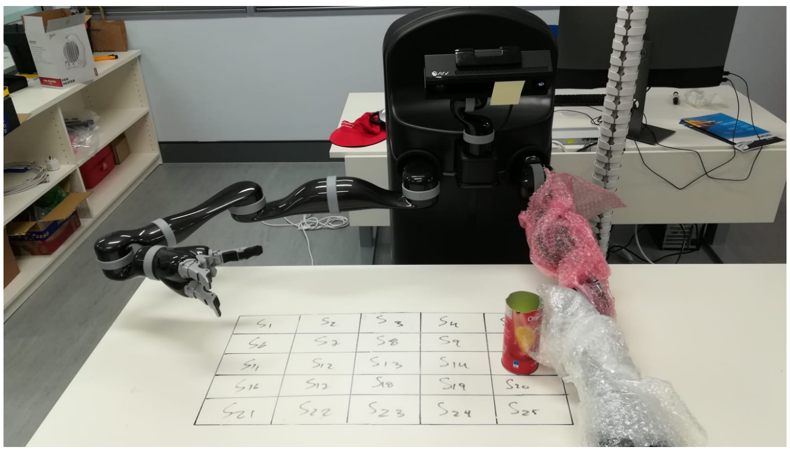


Figure 8: Left arm action on first few steps

Comparing the column values of Table 1 and Table 2, it can be seen that the change in values between using or not using the left arm as a wall, for the first four columns, is not significant relative to the penalty received by a step cost (-50). In contrast, the changes in values in the last column change dramatically indicating the obvious benefits of using the left arm as the object moves closer to the right.

Moreover, experimenting with different parameter values "p" seems to indicate that this markov model method for the rollout strategy can generalize the desired robot use of the left arm for different objects, modelled by different values of p by their relative difficulty for grasping.

The following shows the resulting values from setting $p = 0.5$, which could be used to represent a tennis ball as the target object.

1375.56	1347.52	1284.85	1139.81	781.25
1612.62	1567.42	1469.65	1256.05	781.25
1642.53	1591.67	1483.69	1256.05	781.25
1612.62	1567.42	1469.65	1256.05	781.25
1375.56	1347.52	1284.85	1139.81	781.25

Table 3: Left arm is set as a wall (UP) for p = 0.5

1407.11	1417.63	1445.25	1526.19	1822.59
1664.34	1678.57	1709.19	1773.46	1849.65
1700.89	1714.8	1739.93	1776.8	1849.99
1664.34	1678.57	1709.19	1773.46	1849.65
1407.11	1417.63	1445.25	1526.19	1822.59

Table 4: Left arm is set as a wall (DOWN) for p = 0.5

With a lower probability of grasping, the changes in the values between the two situations, "Wall Set" and "No Wall", become signficant relative to the step penalty cost of putting down the arm from the second or third column, which tells the robot that it would be more benefitial for the robot to use its left arm sooner for more difficult objects.

5.2 Experimental Results

5.2.1 Current Testing

Current results from recent experimentation on simulation and the physical manipulator show that the robot is able to achieve the "desirable behaviour" described on section 1.2.1 of this report. These results show big advancements over the results presented on the interim stage of this project where the not even a successful grasping behaviour was observed.

The differences in results from the interim report stage of this project were due to overall changes in the overall POMDP model, and the use of a different rollout strategy that incorporates more information into the approximation of future expected rewards.

The following shows example scenearios of the desired behaviour consistently seen in experimentation:

A scenario where the robot immediately sees the need to use the left arm on its very first steps due to the belief of the location of the object being close to the edge of the "predefined working region".

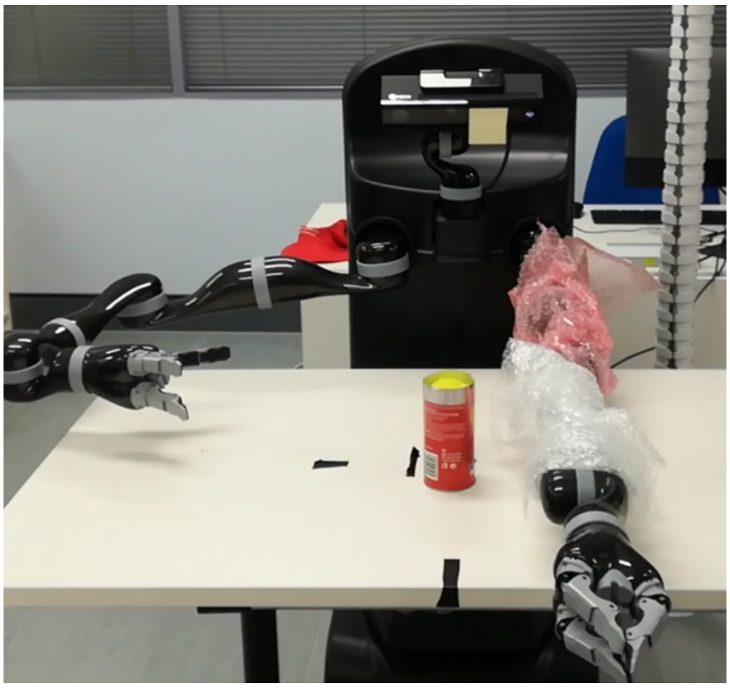


Figure 9: Left arm action on first few steps

A scenario where the experiment was started with the position of the object “relatively” far away from the edge of the “predefined working region” and the robot could prioritize trying to grasp the target object directly, instead of spending extra effort putting down its left arm as a wall when it might not really need it.

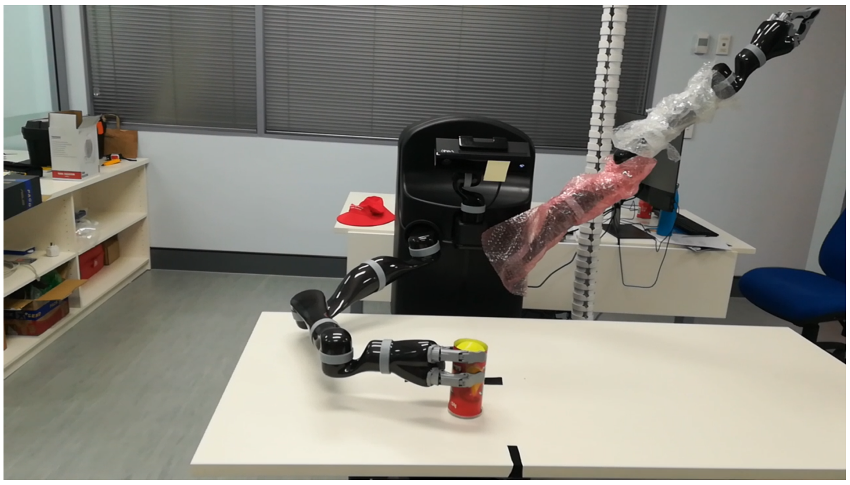


Figure 10: Direct Grasping

A scenario where the robot initially goes to grasp the object directly, but as it gets closer to the object, it suspects that the object might have been pushed closer to the edge of the working area, and decides to use the left arm as a wall before continuing to go for a grasp.

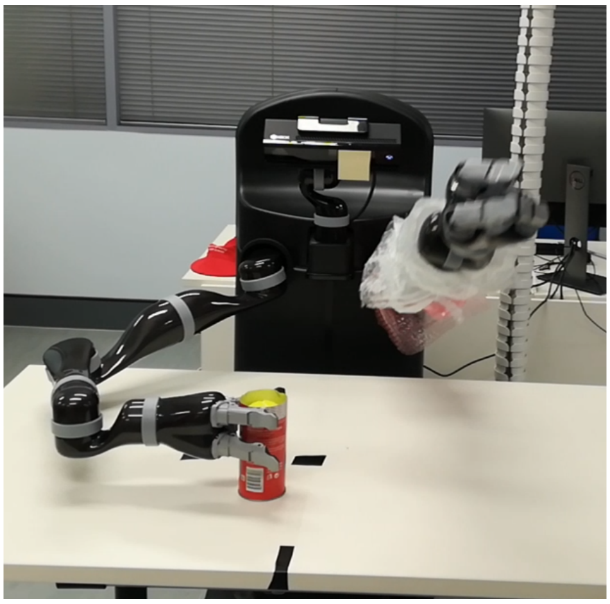


Figure 11: Left arm action after pushing object closer to the edge

5.2.2 Verification of desirable behaviour

To verify that the robot was indeed behaving as desired, and the scenarios similar to the ones presented in 5.2.1 where not events arising by chance, we analyzed the behaviour of the robot during and after every execution.

- **Visual Validation**
  As a simple visual validation of the robot's behaviour, a visual mark was made on the surface of the table to split the working space between a region"A", where the object was considered "far from the edge of the working space" and should not yet use its left arm, and a region "B" where the target object was considered "relatively close" to the edge of working space and the use of the left arm might be handy.

  Simple visual validation showed that the behaviour of the robot was consistent with the "desired behaviour" described on 1.2.1. This is, when the object was initially started on the side of the table where the robot should not use its left arm (region A), it was seen that the robot was always trying to go directly for the grasp. On other hand, when the experiment was started with the object set at the other side of the table (region B), it was seen that the robot performed the "left arm" action on its very first few steps. Moreover, for some cases where the experiment trials were started with the object set on region "A" but close to the threshold mark, it was seen that the robot initially goes for the direct grasping, but as it gets closer to the object, the robot might end up pushing the object onto region "B". For these kind of cases, the robot was seen to put down the left arm when the object was believed to have crossed to region "B".

- **Analysis of the decision making process**
  To further validate the behaviour of the robot, critical information about each experimental run was recorded after each trial. This information included the belief state of the robot, along with a table of the different actions the robot considered, ranked by expected total reward that the robot estimated for each of the actions at its current step. The recorded was information was later used to generate a step by step playback in a simulation environment, and the uncertainty seen by the robot at each step was plotted in a 2D scatterplot.

6 Conclusion

Experiments using the methods described in this report has shown promising results on the robot's ability to determine the value of using its left arm as a wall to facilitate its grasping tasks.

By using its left arm, the robot can enjoy benefits such as uncerainty reduction with respect to the location of the object, avoid scenarios where manipulation tasks becomes impossible due to the object falling out of bounds, and facilitated grasping by using the left arm as a support object while the right arm grasps the target. Analysis and interpretation of the data from uncertainty plots and decision making computations of the scenarios where the robot opts to use the left arm indicate that the behaviour it produces is not due by chance. Instead, the robot is able to weight the cost in time of using its left arm against the possible risks of taking longer time by attempting to approach the object directly, or the object moving out of bounds.

Due to the size of the problem, the search computed by the online POMDP sovler used is not able to explore very deep into the future from its current belief state. As a consequence, the rollout strategy component of the solver seems to plays a very important role for guiding the search towards more promising areas of search. The rollout strategy adopted in this project seems to be a provide good guideance to the robot for seeing the benefits of using the left arm by comparing information about the probabilities that the robot has at any given state of attaining the goal, and the expected rewards obtained when doing so.

Finally, the results (5.1.3) from experimenting with different values of "p" (probability of grasping the target object when the end effector is close enough) for the Markov chain in the rollout strategy, revealed that the methodologies produced in this project might generalize well for testing with other types of objects by modelling the markov chains with different values for its parameter "p" according to the object's difficulty of being grasped.

7 Recommendations

Despite the current results achieved, experiments have only been carried on a pringles tube as a target object. Testing on other objects, especially "more difficult" objects such as a ball or a soap bar must be performed to verify that the methodologies employed so far can scale well. This will be investigated in the remaining days of the duration of this project.

Another thing worth revising is that current Markov chain models only assume a transition in one direction which might not be very representative of general grasping attempts when performing manipulation tasks. If time permits on the remainder of this project's timeframe, this will also be investigated to explore a most robust manner of generating the markov chain approximations in the rollout strategy.

Further limitations from the current methodologies produced in this project is that the left arm is only being used as a wall in one configuration. This design decision was initially made to reduce the complexity of the initial problem with the hope that with little modification or additions to the solutions of the small problem, more general problems could be teackled later on. However, once promising behaviour is seen after testing across different objects, one possible expansion of the current project can be to explore ways to make the robot decide the configuration at which it should put the left arm down in order to maximze uncertainty reduction and aid in its grasping task.

Other possible areas of extension or enhancement from this project include:

- Adding information gathering actions to the POMDP model to enhance the autonomous behaviour of the robot.
- Dynamically learning the model of the robot's surrounding to avoid relying on static abstractions
- Make the robot target manipulation tasks by cooperatively using both of its arms to achieve more challenging tasks such as turning a steering wheel or moving large heavy objects

References

[1] T. Lozano-Perez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *International Journal of Robotics Research*, vol. 3, January 1984.

[2] B. R. Donald, "A geometric approach to error detection and recovery for robot motion planning with uncertainty," *Artificial Intelligence*, vol. 37, no. 1, pp. 223 – 271, 1988.

[3] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1, pp. 99 – 134, 1998.

[4] K. Hsiao, L. P. Kaelbling, and T. Lozano-Perez, "Grasping pomdps," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 4685–4692, April 2007.

[5] K. Hsiao, L. P. Kaelbling, and T. Lozano-Perez, "Robust grasping under object pose uncertainty," *Autonomous Robots: Special Issue on RSS 2010*, 2011.

[6] P. Mons, G. Aleny, and C. Torras, "Pomdp approach to robotized clothes separation," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1324–1329, Oct 2012.

[7] M. Hoerger, J. Song, H. Kurniawati, and A. Elfes, "Pomdp-based candy server: Lessons learned from a seven day demo," in *29th International Conferenceon Automated Planning and Scheduling*, 2019.

[8] J. Pineau, G. J. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for pomdps," pp. 1025–1032, 01 2003.

[9] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," in *Advances in Neural Information Processing Systems 23* (J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, eds.), pp. 2164–2172, Curran Associates, Inc., 2010.

[10] H. Kurniawati and V. Yadav, "An online pomdp solver for uncertainty planning in dynamic environment," in *Proc. Int. Symp. on Robotics Research*, 2013.

[11] K. Seiler, H. Kurniawati, and S. Singh, "An online and approximate solver for pomdps with continuous action space," in *Proc. IEEE Int. Conference on Robotics and Automation (ICRA)*, 2015. Best Conference Paper Award Finalist.

[12] B. Cohen, S. Chitta, and M. Likhachev, "Search-based planning for dual-arm manipulation with upright orientation constraints," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3784–3790, 05 2012.

[13] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, and R. Dillmann, "Humanoid motion planning for dual-arm manipulation and re-grasping tasks," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2464–2470, Oct 2009.

[14] D. Pavlichenko, D. Rodriguez, M. Schwarz, C. Lenz, A. Selvam Periyasamy, and S. Behnke, "Autonomous dual-arm manipulation of familiar objects," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pp. 1–9, Nov 2018.

[15] S. J. Russell and P. Norvig, *Artificial Intelligence - A Modern Approach (3. internat. ed.)*. Pearson Education, 2010.

[16] H. Kurniawati, D. Hsu, and W. S. Lee, "Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces," in *Proc. Robotics: Science and Systems*, 2008.

[17] M. Hoerger, H. Kurniawati, and A. Elfes, "A software framework for planning under partial observability," in *Proc. IEEE/RSJ Int. Conference on Intelligent Robots (IROS)*, 2018.

[18] A. Rastegarpanah, N. Marturi, and R. Stolkin, "Autonomous vision-guided bimanual grasping and manipulation," in *2017 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pp. 1–7, March 2017.

[19] L. Yan, Z. Mu, W. Xu, and B. Yang, "Coordinated compliance control of dual-arm robot for payload manipulation: Master-slave and shared force control," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2697–2702, Oct 2016.

[20] S. S. M. Salehian, N. Figueroa, and A. Billard, "Coordinated multi-arm motion planning: Reaching for moving objects in the face of uncertainty," in *Proceedings of Robotics: Science and Systems*, (AnnArbor, Michigan), June 2016.

[21] A. Yamashita, J. Sasaki, J. Ota, and T. Arai, "Cooperative manipulation of objects by multiple mobile robots with tools," in *Proceedings of the 4th Japan-France/2nd Asia-Europe Congress on Mechatronics*, pp. 310–315, 1998.

[22] A. Stoytchev, "Behavior-grounded representation of tool affordances," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3071–3076, 2005.

[23] Dongheui Lee, Hirotoshi Kunori, and Yoshihiko Nakamura, "Association of whole body motion from tool knowledge for humanoid robots," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2867–2874, Sep. 2008.

[24] A. Xie, F. Ebert, S. Levine, and C. Finn, "Improvisation through physical understanding: Using novel objects as tools with visual foresight," *CoRR*, vol. abs/1904.05538, 2019.

A Plan Structure

A.1 Initial timeline

The timeline structure was included in the initial proposal as an estimated breakdown of the project development timeframe.

- **Weeks 1-5 (14th of January - 17th of February): Review of literature and continued implementation of POMDP Model on simulation**

  The first five weeks are dedicated to the review of knowledge on the area and the continued implementation of the Two-arm POMDP developed on the previous work started on July 2018.

  -*Objectives to be achieved by end of timeframe:* Improve understanding on robotic planning under the presence of uncertainty.
  - *Deliverables to complete by end of timeframe:* First monthly reflection.

- **Weeks 6-8 (18th of February - 10th of March): Familiarizing with MOVO Robot interfaces and Project Proposal**

  These three weeks are dedicated to familiarizing on the required software interfaces to control the mobile manipulator(MOVO) and to develop and submit the project proposal document.

  - *Deliverables to complete by end of timeframe:* Project proposal and second montly reflection.

- **Weeks 9-11 (11th of March - 31st of March): Implementation of POMDP model on MOVO**

  These three weeks are dedicated to integrate the migrate the simulation implementation into control of the physical mobile manipulator.

  -*Objectives to be achieved by end of timeframe:* Acquire skills for implementation of robotic models and simulations under popular robotic frameworks.

- **Weeks 12-16: Testing and Interim Report(1st of April - 5th of May)**
  These five weeks are dedicated to testing and making the necessary modifications on the implementation to try and achieve the proposed manipulation task.

  -*Objectives to be achieved by end of timeframe:* Experimentation of autonomous manipulation task on physical system (MOVO).
  - *Deliverables to complete by end of timeframe:* Interim report and third montly reflection.

- **Weeks 17-20: Further revision and Final Report (6th of May - 2nd of June)**
  These four weeks are dedicated to further testing and revision of the implemenations and algorithms and to the development of the final report.

  - *Deliverables to complete by end of timeframe:* Fourth reflection.

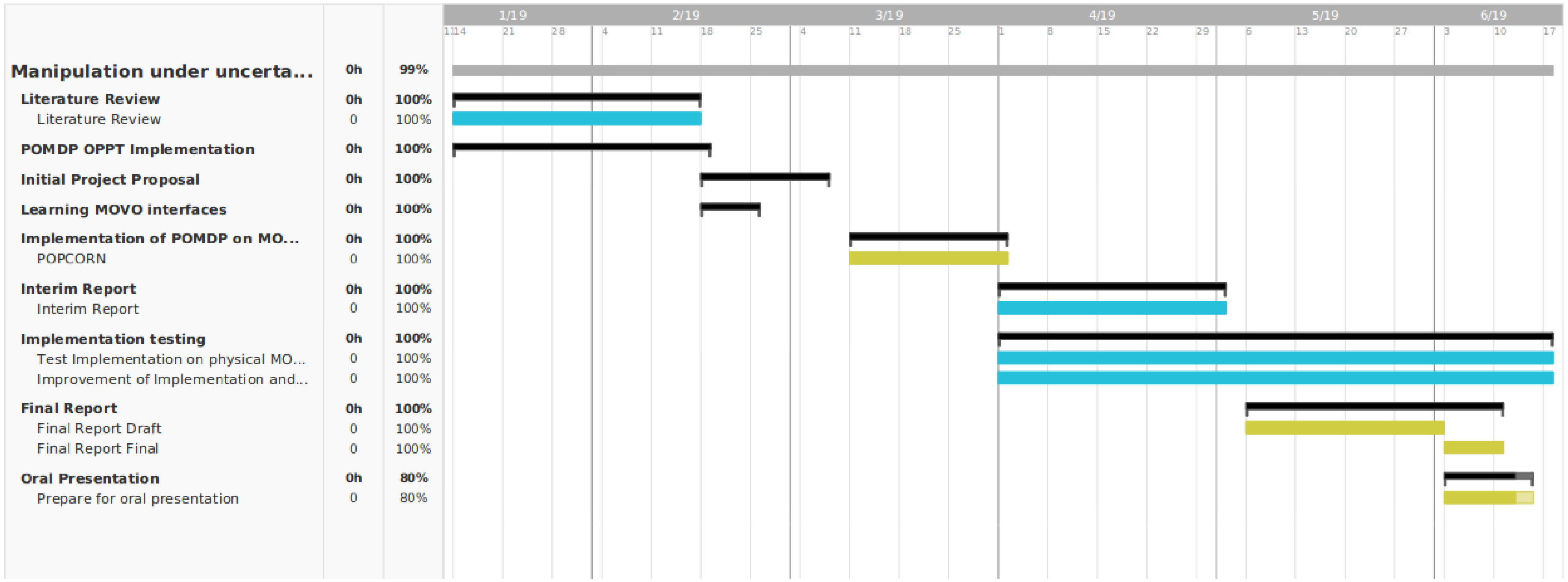
- **Weeks 21-22: Final Report Submission and Oral Presentation (3rd of June - 17th of June)**
  These two weeks are dedicated to final revisions on the report, preparting for the oral presentation, and mantaining the implementations for the final products.

  - *Deliverables to complete by end of timeframe:* Fifth reflection, final report, and oral presentation.

- **Weeks 23-24: Final Report Submission and Oral Presentation (17th of June - 30th of June)**
  Since, the final assessments are to be submitted with two weeks of anticipation before the actual end of the placement, the remaining weeks, will be dedicated to fixing and managing any remaining work or improvement that can be made.

The following figure shows the updated Gantt Chart of the project development.

A.2 Gantt Chart



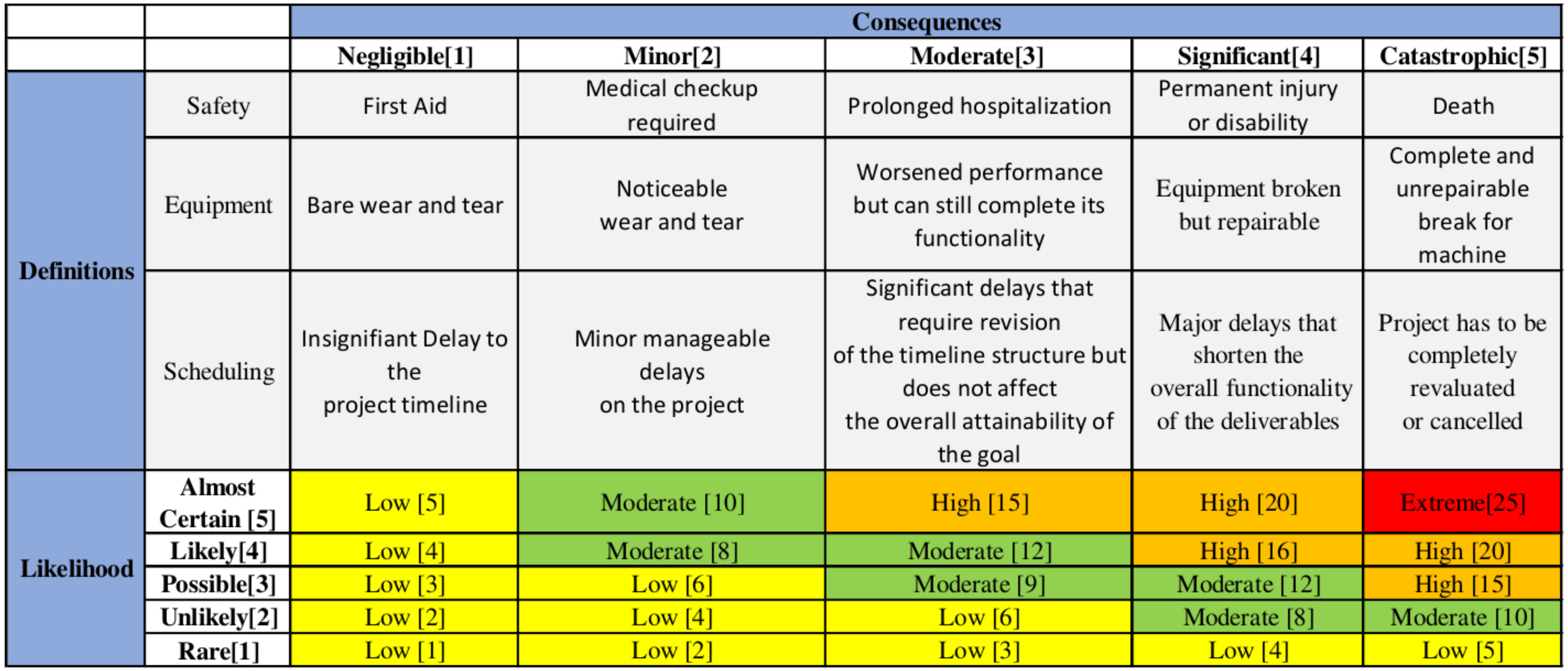
Manipulation under uncerta...	**0h**	**99%**
Literature Review	**0h**	**100%**
Literature Review	0	100%
POMDP OPPT Implementation	**0h**	**100%**
Initial Project Proposal	**0h**	**100%**
Learning MOVO interfaces	**0h**	**100%**
Implementation of POMDP on MO...	**0h**	**100%**
POPCORN	0	100%
Interim Report	**0h**	**100%**
Interim Report	0	100%
Implementation testing	**0h**	**100%**
Test Implementation on physical MO...	0	100%
Improvement of Implementation and...	0	100%
Final Report	**0h**	**100%**
Final Report Draft	0	100%
Final Report Final	0	100%
Oral Presentation	**0h**	**80%**
Prepare for oral presentation	0	80%

Figure 12: Gantt Chart for project management

A.3 Risk Assesment

Since all stages for the development of the project are carried out inside a controlled laboratory environment, and most of the work to be done is software related, the risk factors to consider are still the same as the ones detailed in the initial project proposal. The tables below provide the relevant information concerning possible risks, and the controls that will be adopted to avoid and mitigate them. The severity of this risks are defined relative to the risk matrix presented in figure 4.

A.3.1 Risk Matrix



		Consequences				
		Negligible[1]	**Minor[2]**	**Moderate[3]**	**Significant[4]**	**Catastrophic[5]**
Definitions	Safety	First Aid	Medical checkup required	Prolonged hospitalization	Permanent injury or disability	Death
	Equipment	Bare wear and tear	Noticeable wear and tear	Worsened performance but can still complete its functionality	Equipment broken but repairable	Complete and unrepairable break for machine
	Scheduling	Insignifiant Delay to the project timeline	Minor manageable delays on the project	Significant delays that require revision of the timeline structure but does not affect the overall attainability of the goal	Major delays that shorten the overall functionality of the deliverables	Project has to be completely revaluated or cancelled
Likelihood	**Almost Certain [5]**	Low [5]	Moderate [10]	High [15]	High [20]	Extreme[25]
	Likely[4]	Low [4]	Moderate [8]	Moderate [12]	High [16]	High [20]
	Possible[3]	Low [3]	Low [6]	Moderate [9]	Moderate [12]	High [15]
	Unlikely[2]	Low [2]	Low [4]	Low [6]	Moderate [8]	Moderate [10]
	Rare[1]	Low [1]	Low [2]	Low [3]	Low [4]	Low [5]

Figure 13: Risk matrix for project control

A.3.2 Safety risks

Major safety risks associated from a software laboratory controlled working environment are very low. The following table summarizes some of the possible risk factors.

Activity	Associated Risk	Risk Control Method	Residual Risk
Sitting in a bad posture for long hours	Physical pain and injury (LOW)	Adopt a good posture while working. Standing desks are also available to alternate the working stance	LOW
Working for extended hours in front of the computer without a break	Eye stress and dryness (LOW)	Take regular breaks and blink every 20 minutes to hydrate the eyes	LOW

Table 5: Safety risks table

A.3.3 Equipment risks

Equipment risks are more likely than safety risks for this project are some of the resources are very expensive like the mobile manipulator. The following table summarizes some of the possible risk factors.

Activity	Associated Risk	Risk Control Method	Residual Risk
Unproper use of MOVO	Damage to the system, with the possibility of permanently damaging internal components (HIGH)	Get inducted on the proper use of the manipulator and make sure to review the manual when in doubt.	LOW
Electrical failures and resets	Power surges can damaged the connected electrical components in the labora-tory(MODERATE)	Ensure all equipment used is electrically tested and tagged. Ensure power switches are OFF when not using the associated equipment	LOW

Table 6: Equipment risks table

A.3.4 Scheduling risks

The following table summarizes some of the possible risk factors that might affect the proposed timeline structure of this project:

Activity	Associated Risk	Risk Control Method	Residual Risk
Breakdown or malfunction of physical system (MOVO)	Delays in the implementation and testing phases of the project (MODERATE)	Ensure appropiate use of the physical system via controls detailed in the equipment risks section.	LOW
Misunderstanding implementations requirements and objectives(LOW)	Waste of project development time and reimplementation required (MODERATE)	Revise implementation requirements for software modules with the relevant supervisor beforehand	LOW

Table 7: Scheduling risks table

A.3.5 Project Opportunities

The following lists some of the opportunities for the project that might present along its development:

- If novel algorithms relevant to the implementation of the project are published in the literature, the existing implementation could see improved performance by attempting to integrate the new methods.
- If the project is successfully completed ahead of schedule, extending functionalities could be added to the mobile manipulator.
- If the project is completed with great success, the implementation can be use a base work for further research and enhancements.

B Professional Development

As of the writing date of this report, all monthly reflections have been submitted. The following presents the SEAL analysis of some key lessons and skills learned throughout my project and the corresponding EA Stage 1 Competencies covered.

B.1 SEAL Analysis February : Use of new software tools and frameworks

EA Stage One Competencies covered: 1.1-1.4, 2.1, 2.3, 3.3

B.1.1 Situation:

To start with the simulation and execution of the theoretical framework designed for the research, I had to start familiarizing with the proper tools and software engines such as GAZEBO (simulation engine) and ROS (Robotic Operating System) for implementation. I was reluctant to ask for help as I believed I could somehow figure it out by myself.

B.1.2 Effect:

I was very stressed and overwhelmed by the amount of stuff to be learned and did not know how to begin. I felt like I did not deserved to be offered such a great placement opportunity if I did not managed to complete the learning by myself.

B.1.3 Action:

I decided to communicate my struggles with my supervisor. She was very understanding and offered me a learning path to follow. This made the learning so much easier and fun.

B.1.4 Learning:

From this, I learned that it is always fine to ask for help. Even if the struggle might seem minimal, you can save a lot of time by simply asking instead of trying to avoid bothering your peers and supervisors.

B.2 SEAL Analysis March : Site visits from course coordinator and staff

EA Stage One Competencies covered: 3.2, 3.3, 3.5

B.2.1 Situation:

On February the 22 nd I had an online meeting along my project supervisor with the course coordinator and staff. The purpose of this meeting was to see how I was adapting to my working environment and to make sure that I had a good understanding and scope on the project I was working on. When asked about the details of the project, instead of describing it in a brief and clear manner, I got the impression that I started babbling about random details and theory about the project without any cohesion between what I was talking about.

B.2.2 Effect:

This catch-up meeting made me realise there were still some missing gaps and details on the project that I needed clarified. Moreover, I also noted that I have to work on my oral communication and presentation skills to make sure situations like this do not occur in the future.

B.2.3 Action:

I sat down with my supervisors to clarify some details and things I needed to improve, including my communication skills.

B.2.4 Learning:

I learnt although I might think I have a crystal-clear understanding on something or I am can handle something easily, there is always room for improvement.

B.3 SEAL Analysis April : Erratic behaviour of the Manipulation System

EA Stage One Competencies covered: 1.1-1.3, 2.1-2.2

B.3.1 Situation:

The mobile manipulator comes with a demo program that runs some of the basic movements that the robot can perform and can also help to check that movement control through software is functioning properly. Last week, this demo program stopped working without any change to its software, which indicated that something was not right with the physical system

B.3.2 Effect:

The erratic behaviour on the robot, meant that testing some of the desired functionalities had to be suspended until the problem was resolved. Therefore, focus was shifted into debugging the behaviour of the robot.

B.3.3 Action:

Initially, the simplest explanation of the sudden change in behaviour was that some sensors might have experienced tear and wear in the system. However, I started tracing back and thinking on the possible changes that we might have missed and could have caused the erratic behaviour.This debugging process required some deep examination of the system,since the cause of the strange behaviour could have been from either the software side, the hardware side, or both.
After spending some time studying the system, the problem was found to be an error on my understanding on how a component of the system was supposed to behave compared to its actual functionality.

B.3.4 Learning:

From this experience, I learned that because hardware systems are not always reliable, it is often easier to just blame the product, rather than taking the time to deeply focus on the possible sources of error. However, errors must be traced and deeply examined until a definite conclusion of the faults can be made.

B.4 SEAL Analysis May : Communication Skills

EA Stage One Competencies covered: 3.2, 3.5

B.4.1 Situation:

Recent meetings with my supervisors have made me realized that my communication skills are still not very good. When trying to explain a problem or an issue, I still find myself circling around or going off topic instead of directly addressing the point.

B.4.2 Effect:

I started reflecting on the possible problems I might have with communication. Initially, I thought that my difficulty presenting information was due to the English not being my first language, so thoughts I had in my native language sometimes did not make sense in a direct translation to English. However, I have realized my main problem is on synthesizing the information I want to convey.

B.4.3 Action:

As an effort to improve my communication skills, I started practicing both my oral communication and information synthesis in my free time.

B.4.4 Learning:

From this experience, I learned that no matter how much knowledge you can acquire, the information you hold is not of much use if you cant communicate it nicely and clearly. These kinds of soft skills are often ignored to prioritize more technical skills. However, I have found that they might be just as important.

B.5 SEAL Analysis June : Final Deadlines

EA Stage One Competencies covered: 3.3, 3.5

B.5.1 Situation:

As a requirement for an on-time graduation at the end of the semester, I was asked to present my final report a few weeks ahead some of my classmates. At the start of the project, I have proposed myself to try and accomplish good results with this stricter timeline, to be able to include them in the final report. This turned out to be more challenging than expected.

B.5.2 Effect:

Despite keeping up with the initially proposed work timeline, with the deadlines closing in, it was harder to balance work on both, improving results and managing my university assignments at the same time.

B.5.3 Action:

To achieve what initially proposed, I decided to split my time into working on the project during the day and putting extra hours of work at night for my university assignments. Moreover, I extended my working time to include weekends as well. After a few shifts of this, my workload eventually lightened, and I was able to bring back my work schedule back to normal.

B.5.4 Learning:

Apart from becoming more responsible and organized with my time, I learned that initial proposed timelines can be more optimistic when they are first formulated than what they really are. This is where having a realistic project scope and being able to anticipate the work breakdown structure with great detail, can make a big difference.

B.6 Summary of EA Stage One Competencies

The following list summarises the EA Stage One Competencies covered so far through the development of the project. This list includes the competencies discussed in the SEAL analysis above, and the extra events indicated in the monthly reflective journals submitted.

- 1.1. Comprehensive,theory based understanding of the underpinning natural and physical sciences and the engineering fundamentals applicable to the engineering discipline.
- 1.2. Conceptual understanding of the mathematics, numerical analysis, statistics, and computer and information sciences which underpin the engineering discipline.
- 1.3. In-depth understanding of specialist bodies of knowledge within the engineering discipline.
- 1.4. Discernment of knowledge development and research directions within the engineering discipline
- 1.5. Knowledge of engineering design practice and contextual factors impacting the engineering discipline.
- 1.6. Understanding of the scope, principles, norms, accountabilities and bounds of sustainable engineering practice in the specific discipline.
- 2.1. Application of established engineering methods to complex engineering problem solving.
- 2.2. Fluent application of engineering techniques, tools and resources.
- 2.3. Application of systematic engineering synthesis and design processes.
- 2.4. Application of systematic approaches to the conduct and management of engineering projects.
- 3.1. Ethical conduct and professional accountability
- 3.2. Effective oral and written communication in professional and lay domains.
- 3.3. Creative, innovative and pro-active demeanour
- 3.4. Professional use and management of information.
- 3.5.Orderly management of self, and professional conduct.
- 3.6 Effective team membership and team leadership.