Vision-Based Autonomous Robotic Manipulation
Learning from Observation and Exploration *

STAGE 2 PROPOSAL

AUTHOR:
Fangyi Zhang
fangyi.zhang@hdr.qut.edu.au
Student ID: 09314181

PRINCIPLE SUPERVISOR:
Professor Peter Corke

ASSOCIATE SUPERVISORS:
Dr Jürgen Leitner
Associate Professor Ben Upcroft

ARC Centre of Excellence for Robotic Vision
School of Electrical Engineering and Computer Science
Science and Engineering Faculty
Queensland University of Technology

July 2, 2015

*This research is sponsored by the ARC Centre of Excellence for Robotic Vision and the Queensland University of Technology, in the form of ARC Centre of Excellence for Robotics Scholarship and QUT HDR Tuition Fee Sponsorship, respectively.
Contents

1 PROPOSED THESIS TITLE AND TYPE ................................................................. 1

2 PROPOSED SUPERVISORY TEAM AND THEIR CREDENTIALS ................................ 1

3 BACKGROUND AND LITERATURE REVIEW ......................................................... 1

3.1 Introductory Statement ..................................................................................... 1

3.2 Literature Review ............................................................................................ 2

3.2.1 Vision-based Robotic Manipulation ................................................................ 2

3.2.2 Reinforcement Learning ............................................................................. 3

3.2.3 Q-learning .................................................................................................. 4

3.2.4 Deep Reinforcement Learning ..................................................................... 4

3.2.5 Safe Exploration in Reinforcement Learning ............................................. 5

3.2.6 Learning from Demonstration ..................................................................... 6

3.2.7 Learning Manipulation Action Plans from Youtube Videos ......................... 7

3.2.8 Apprenticeship Learning ............................................................................ 7

3.2.9 Summary .................................................................................................... 8

3.3 Research Problem ........................................................................................... 8

4 PROGRAM AND DESIGN OF THE RESEARCH INVESTIGATION ......................... 9

4.1 Objectives, Methodology and Research Plan ................................................. 9

4.1.1 Visual Perception Method ......................................................................... 9

4.1.2 Learning from Observation ........................................................................ 9

4.1.3 Learning from Exploration ....................................................................... 9

4.1.4 Safety Guarantee ....................................................................................... 10

4.2 Resources and Funding Required ................................................................... 10

4.3 Individual Contribution to the Research Team .............................................. 10

4.4 Timeline for Completion ............................................................................... 11

5 REFERENCE LIST .............................................................................................. 11

6 APPENDIX .......................................................................................................... 13

6.1 Research Ethics Integrity and Safety (REIS) .................................................... 13

6.2 Coursework ..................................................................................................... 13

6.3 Ethical Clearance Statement .......................................................................... 14

6.4 Intellectual Property Statement ..................................................................... 14

6.5 Health and Safety Statement .......................................................................... 14

6.6 Collaborative Arrangement Statement .......................................................... 14

6.7 Memorandum of Understanding for External Supervisors Statement .......... 14

6.8 Data Management Statement ....................................................................... 14
1 PROPOSED THESIS TITLE AND TYPE

• Thesis Title: Vision-Based Autonomous Robotic Manipulation Learning from Observation and Exploration

• Thesis Type: Traditional Thesis by Monograph (Chapters)

2 PROPOSED SUPERVISORY TEAM AND THEIR CREDENTIALS

• Principle Supervisor: Professor Peter Corke

  Peter is a Professor at the Queensland University of Technology (QUT) and the director of the ARC Centre of Excellence for Robotic Vision (ACRV). He is known for his research in vision-based robot control, field robotics and wireless sensor networks. He received B.Eng and M.Eng.Sc. degrees, both in Electrical Engineering, and a PhD in Mechanical and Manufacturing Engineering, all from the University of Melbourne. Prior to QUT he was a senior principal research scientist at CSIRO.

• Associate Supervisor: Dr Jürgen Leitner

  Jürgen is a researcher at ACRV. His research is on Robotics and Computer Vision looking at learning and Artificial Intelligence (AI). Prior to ACRV he worked at the IDSIA Robotics Lab and received a PhD from the Università della Svizzera Italiana (USI). Previously he worked at the Advanced Concepts Team of the European Space Agency. He studied Space Robotics in a Joint European Master Programme (SpaceMaster) at Aalto (TKK) and the Kiruna Space Campus (LTU).

• Associate Supervisor: Associate Professor Ben Upcroft

  Ben is an Associate Professor at QUT. His interests lie in the development of vision systems for long term robotic applications ranging from underwater and ground to airborne autonomous platforms. The overarching theme of his work is in robust visual perception for robotics. Prior to QUT he was a Senior Lecturer at University of Queensland. He received a B.Sc. degree and PhD both in Physics, from University of Queensland.

3 BACKGROUND AND LITERATURE REVIEW

3.1 Introductory Statement

Robots are widely used in industry, such as welding robots, material handling robots, painting robots, palletizing robots, and assembly robots. They have highly increased manufacturing productivity. However, the number of robots successfully adopted in the environments where humans live (daily life environments) is quite limited, because, unlike highly controlled industrial environments, daily life environments are highly dynamic and can hardly be controlled.

In industry, the environments can be perfectly arranged for robots to implement manipulation tasks, such as specifically deployed objects, simplified background colours for visual
perception, and separated areas with no humans. However, this kind of arrangement is impractical in daily life environments where objects’ positions vary quite often; environments’ backgrounds are complex; human behaviours are dynamic and hard to predict. In such uncontrolled environments, it is challenging for robots to do houseworks, cook meals, and clean streets, which consist of series of manipulation tasks.

Nevertheless, humans implement manipulation tasks well in daily life environments. Inspired by this, robots should be potentially more capable to manipulate in daily life environments to help humans more. Considering that humans gain manipulation skills by learning from other expertises and improving after mistakes, this research aims at enabling robots to understand manipulation tasks from observing how others do through vision (learning from observation) and master manipulation skills through trial and error (learning from exploration). In addition, to guarantee the safety in the learning process, feasible and safe exploration mechanisms will also be studied.

3.2 Literature Review

There is much previous literature relevant to vision-based robotic manipulation and learning from observation and exploration, especially reinforcement learning, learning from demonstration, and their intersection research.

3.2.1 Vision-based Robotic Manipulation

Vision-based robotic manipulation is the process by which robots use their manipulators (such as robotic arms) to rearrange environments [1], based on visual perception. The early vision-based robotic manipulation was implemented using pose-based (position and orientation) closed-loop control, where vision was typically used to extract the pose of an object as an input for a manipulation controller at the beginning of a task [2]. Normally, the controller was based on kinematic models without vision information in the control loop. The applicable scale of this control method is quite small, since it makes sense only when kinematic models are precise and environments stay unchanged, unfortunately the real world changes in most cases.

In contrast, vision-based closed-loop control methods can continuously adapt to dynamic environments [3]. However, the computation cost of vision-based closed-loop methods is higher than that of pose-based closed-loop methods, since image processing is necessary in each cycle. Most current vision-based robotic manipulation methods are closed-loop based on visual perception. A vision-based manipulation system was implemented on a Johns Hopkins “Steady Hand Robot” for cooperative manipulation at millimeter to micrometer scales, using virtual fixtures [4]. With both monocular and binocular vision cues, various closed-loop visual strategies were applied to enable robots to manipulate both known and unknown objects [5].

Until now, various learning methods have been applied to implement complex manipulation tasks in the real world. For example, with continuous hidden Markov models (HMMs), a humanoid robot was able to learn dual-arm manipulation tasks from human demonstrations through vision [6]. However, in different manipulation tasks, learning algorithms vary a lot. A specific algorithm is effective for only one specific task. More study is necessary to make vision-based robotic manipulation learning more generalized and autonomous, so that robots can adapt to specific tasks within the same broad class of a learned task.
3.2.2 Reinforcement Learning

Reinforcement learning (RL) is a method of iteratively learning from trial and error. It is widely used and studied in the areas of machine learning, optimal control, and robotics. The modern field of reinforcement learning arose in the late 1980s [7]. In the robotics area, RL is used to enable robots to both acquire new skills and improve the performance of mastered skills through experience.

Figure 1 shows how a RL algorithm (the agent) and the environment interact at each time step, \( t = 0, 1, 2, 3, \ldots \) [7]. At each time step \( t \), based on the received reward \( r_t \in \mathbb{R} \) and representation of the environment’s state \( s_t \in S \), the agent selects an action \( a_t \in A \) to interact with the environment. \( S \) denotes the set of possible states; \( A \) is the set of available actions. One time step later, in part as a consequence of the action, the agent receives a new reward \( r_{t+1} \in \mathbb{R} \), and ends up in a new state \( s_{t+1} \in S \). The agent’s goal is to learn a policy \( (\pi : S \rightarrow A) \), that maps a current state to an action, to maximize the long-term reward it receives.

Formally, RL can be defined as a Markov Decision Process (MDP). It is represented by a 4-tuple, \( M = \{S, A, p, r\} \) [8]. Function \( r : S \times A \rightarrow \mathbb{R} \) is a reward function which calculates immediate reward \( r_{t+1} \) resulted in by taking action \( a_t \) in state \( s_t \). In some cases, the immediate reward is directly provided by the environment. For example, when playing a video game, the immediate reward is the earned scores directly provided by the video game engine, where the reward function is in the game engine. In the other cases, the reward function is included in the agent. Function \( p : S \times A \rightarrow \Delta \) is a transition function, where \( \Delta \) is a set of probability distributions and specifies the probability of ending up in state \( \hat{s}_{t+1} \) when performing action \( a_t \) in state \( s_t \). \( \hat{s}_{t+1} \) is the state predicted by the transition function, \( s_{t+1} \) in Figure 1 is the true state. The transition function is optional and included in the agent. Its output \( \Delta \) is part of the input of a value function which estimates the long-term reward to guide the policy determination. Generally, RL learns the value function.

RL for robotic applications has been verified by much previous research [9, 10]. For example, with State-Action-Reward-State-Action (SARSA) learning algorithm, a Mitsubishi Pa10 robot successfully learned to play ball-in-a-cup [11]. By applying value function approximation techniques for goal learning and direct policy search methods for shape learning, a learning-to-pour-liquid task was implemented in simulations as well as using a Pa10 robot [12].

![Figure 1: The agent-environment interaction in reinforcement learning](image)
3.2.3 Q-learning

A widely used RL algorithm is Q-learning which needs no transition function (model-free) [13]. It is a Temporal-Difference (TD) control algorithm [7]. With Q-learning, an AIBO robot was able to learn to walk on new surfaces [14]. By using a Q-learning algorithm accelerated through memory-based sweeping and “adjoining property” enforcing, a docking task was learned in less than 1 hour on a vision-based mobile robot (PeopleBot) [15]. A mobile robot learned a navigation task from scratch in around 20 minutes through Q-learning [16].

In its simplest form, one-step Q-learning is defined by

\[
Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t \left[ r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]
\]

(1)

where \( Q \) is the value function (Q-value) which calculates the quantitative quality of a state-action pairing; \( \gamma \) (\( 0 \leq \gamma < 1 \)) is the discount factor which trades off the importance of history rewards; \( \alpha_t \) (\( 0 \leq \alpha_t < 1 \)) is the learning rate [7]. Q-learning algorithm is shown in Algorithm 1. Normally \( \alpha_t \) decays over time for the learning to converge. The algorithm chooses an action for each time step through \( \arg \max_{a_t} Q(s_t, a_t) \).

Algorithm 1: The Q-learning algorithm

1. Initialize \( Q(s_t, a_t) \) arbitrarily
2. Repeat (for each episode):
   3. Initialize \( s_t \)
   4. Repeat (for each step of episode, \( t \)):
      5. Choose \( a_t \) in \( s_t \) according to \( \max_{a_t} Q(s_t, a_t) \)
      6. Take action \( a_t \), observe \( r_{t+1}, s_{t+1} \)
      7. \( Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t \left[ r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \)
   8. \( s_t \leftarrow s_{t+1} \)
9. until \( s_t \) is terminal

Because of the absence of a transition function, Q-learning provides a good solution for the cases where the modelling of a transition function is too complex and impractical. However, an appropriate exploitation and exploration strategy (e.g., \( \varepsilon \)-greedy [7]) is necessary to make sure Q-learning can eventually find out the optimal policy, otherwise the exploration might be limited in a sub-optimal interval by \( \max_{a_t} Q(s_t, a_t) \). For practical applications, more specific designs for state representation, action representation, and an approximate model are needed. The exact performance of specific algorithms depends a lot on these further designs.

3.2.4 Deep Reinforcement Learning

A type of RL algorithm which uses a deep neural network (DNN) to approximate a value function is called deep reinforcement learning (DRL). By combining Q-learning and a deep convolutional neural network (CNN) [17], a human-level of control for Atari 2600 games was implemented [18]. The method is called Deep Q-network (DQN), a specific DRL algorithm. It worked at a level that was comparable to that of a professional human games tester across the set of 49 games tested.
DQN is an end-to-end algorithm which directly maps raw pixel inputs to action commands. The inputs of DQN include raw pixels of game video frames, received scores and the number of available actions in each game. No pre-input feature extraction is needed, which is necessary for most previous RL algorithms and neural networks. The end-to-end architecture made it possible to play different games without any modifications. The only one thing was to let the algorithm improve policies through playing games over and over again.

In the implementation of DQN, in addition to using a convolutional network to estimate Q-values, an experience replay mechanism [19] was used to randomly sample previous records to increase the learning efficiency. However, the training process still took around 38 days to achieve the comparable human-level performance for 49 games. Although the training time-cost is still too high for real robotic applications, with the fast development of computation devices, it can be reduced to an acceptable scale in the near future.

In the training process of DQN, the game engine worked as a reward function, but for real robotic applications, no such engine exists. To apply DQN in real robotic applications, a reward function is needed to assess trials. Besides, more study is necessary for noisy and more complicated real-world settings. In addition, the aforementioned DQN conducted the training for each Atari 2600 game independently. Learning based on the learned results from other games might be another good way to increase learning efficiency. Moreover, when using DRL in robotic applications, some other kinds of RL algorithms and DNNs can be considered according to specific requirements.

3.2.5 Safe Exploration in Reinforcement Learning

To apply RL in robotics, the safety of exploration is a key issue of the learning process in the real world [20]. The behaviours of robots under the control of RL algorithms are uncertain, some of which might be harmful to humans, robots themselves, and the environments around. An efficient algorithm for guaranteed safety was developed through applying some constraints to restrict attention to a subset of guaranteed safe policies which was provided by humans [20]. However, the exploration under the restriction of this algorithm might be suboptimal. More study is necessary to trade off the balance between safety and optimization.

Another possible solution is to conduct the learning process in simulation environments where no damage will be resulted in to the real world. In addition, exploration in simulation can be speeded up faster than real time. However, the difference between simulation environments and the real world is inevitable, i.e, the difference between simulation models and true systems. Although some previous research provides some methods to deal with the difference, and to some extent verifies the effectiveness of transferring policies learned in simulation to the real world, completely getting rid of the difference is impractical. If the difference is beneficial for the reward received in simulation, RL algorithms will take into account such difference. As a consequence, the learned policies might work well in simulation but poorly in the real world [21].

To better trade off the effectiveness of learned skills and the safety of exploration, a better solution could be to first conduct exploration in simulation environments, then fine-tune in the real world (balanced solution). Through the balanced solution, soccer robots were able to learn to cooperate with each other [22]. The balanced solution can not only guarantee the exploration safety but also ensure the effectiveness of learned skills in the real world.

Regarding simulation environments, some standard platforms can be used to simplify their
construction process. One is V-REP, a versatile, scalable, yet powerful general-purpose robot simulation framework [23]. V-REP can be used for rapid algorithm development and prototyping. It has been widely used in the academic as well as industrial field. Another one is the SL framework [24], which allows users to switch between a simulated robot and a real one with a simple recompile. In addition, ROS [25], a robot operating system which is compatible with most simulation platforms and real robots, can be used to construct the software framework.

3.2.6 Learning from Demonstration

Learning from demonstration (LfD) is another widely used learning method which enables robots to learn actions. It approximates a map between states and actions from teacher’s demonstrations. A LfD framework consists of a policy derivation unit and a policy execution unit [26], as shown in Figure 2. In the policy derivation unit, the teacher’s demonstrations include all states and selected actions throughout a teacher execution of a task. A demonstration $d_j \in D$ is presented as $k_j$ pairs of observations and actions: $d_j = (z_j^i, a_j^i)$, $z_j^i \in Z$, $a_j^i \in A$, $i = 0 \cdots k_j$. According to the demonstration dataset $D$, the policy $\pi : Z \rightarrow A$ is constructed by a policy derivation algorithm, i.e., extracting the relation between states $Z$ and actions $A$. Based on the policy $\pi$, the policy execution unit takes action $a$ according to perceived state $z$, to interact with the world. $z$ is the perception result of the true world state $s \epsilon S$. The mapping $M : S \rightarrow Z$ between $S$ and $Z$ is determined by a perception algorithm.

There are various demonstration gathering approaches. Teleoperation was successfully used for demonstrations in the learning of a controller for sustained inverted flight on an autonomous helicopter [27], robot kicking motions [28], and object grasping [29]. A human assistance method was used in the kinesthetic teaching of a humanoid robot, where the joints of the robot are passively moved through desired motions by humans [30]. Motion capture systems were successfully used to teach human motion [31] and manipulation tasks [32].

All these demonstration gathering methods are effective in practical applications. However, due to the existence of too much accessory equipment and sensors, the implementation is still
too complicated and inconvenient, sometimes impractical. A more practical approach is that robots learn from human-performed demonstrations through their on-body cameras, without other accessory sensors. More study needs to be conducted for this approach.

Regarding policy derivation methods, there are three major types, i.e., mapping functions, system models, and execution plans [26]. A Bayesian likelihood method was used to approximate actions selection for a humanoid robot in a button pressing task [33]. Support Vector Machines (SVMs) were used to classify behaviours in a robotic ball sorting task [34]. Using a plan-based method, a humanoid learned a generalized plan for a repetitive ball collection task, from only two demonstrations [35]. A system model based LfD method was successfully used to optimize a neural network controller for autonomous helicopter flight [36].

Among these three types of policy derivation methods, compared with mapping function based methods, the execution plan based methods converge faster, because of the existence of a plan frame which constrains the learning process to a smaller range. However, this constraint also limits the application scale of execution plan based methods.

### 3.2.7 Learning Manipulation Action Plans from Youtube Videos

A particular type of LfD algorithm, whose demonstrations are unconstrained Youtube videos (a fully-labelled cooking dataset, YouCook [37]), was proposed for robot to learn manipulation action plans [38]. It consisted of a probabilistic-manipulation-action-grammar based parsing module and two CNN based recognition modules: one for grasping type recognition; the other for object recognition. The parsing module was to generate visual sentences for robot manipulation. The algorithm was able to recognize and generate action plans robustly. It achieved an overall recognition accuracy of 79% on objects, of 91% on grasping types, and of 83% on predicted actions. The overall action commands accuracy was 68%.

According to its performance, this algorithm is promising for further improvements and to be applied on real robots. However, it is a supervised learning method which needs fully labelled data. Developing an unsupervised learning algorithm would make the learning process independent to the data labelling.

### 3.2.8 Apprenticeship Learning

Apprenticeship learning [39] is the intersection of RL and LfD. It uses inverse reinforcement learning (IRL) [40] to recover an unknown reward function through learning from demonstrations. Using apprenticeship learning, an algorithm was able to learn different driving styles in a car driving simulation platform, where the algorithm constructed the reward function through linear regression [39]. Apprenticeship learning was also used in the control of an aerobatic helicopter flight, where it outperformed previous state-of-the-art approaches, in both the flipping and rolling tasks [41].

Apprenticeship learning is able to remedy the weaknesses of both RL and LfD. LfD replaces the design of a reward function. RL enlarges the adaptability of LfD, because of its self-improvable ability during exploration. Apprenticeship learning is quite suitable for the problems where the reward function is unclear or too complex to be hand designed. In the implementation, other than linear regression, many other function approximation methods should also be considered, such as DNNs. A more generalized function approximation mechanism might make apprenticeship learning more powerful.
3.2.9 Summary

RL is an effective method to enable robots to autonomously learn some skills through exploration. Its effectiveness has been verified in a large number of robotic applications. Q-learning, a model-free RL algorithm, which significantly simplifies the modelling process, is the most widely used form of RL.

DQN, a DRL algorithm, which combined Q-learning and CNN, achieved an excellent performance in playing Atari 2600 games. It was able to autonomously learn the game tricks through iteratively playing the game (exploration), without other prior knowledge other than the number of available inputs. It directly learned from raw pixel information, no pre-input feature extraction is necessary. The algorithm was able to be applied to different Atari 2600 games without any modifications.

However, in the area of real robotic manipulation, no such powerful and generalized method exists. DQN is a good reference to enable robots to autonomously learn manipulation skills from exploration. To apply DQN on real robots, or to develop a new DRL algorithm for real robotic manipulation, a general reward function is needed. The reward function should be able to be used to assess the exploration for different kinds of manipulation tasks. Inspired by apprenticeship learning and the method of learning manipulation action plans from Youtube videos, we can use LfD to learn the general reward function from observing demonstrations, i.e., learn from observation.

Regarding the exploration safety, the mechanism introduced in Section 3.2.5, first conducting exploration in simulation environments, then fine-tuning in the real world, can be used to not only guarantee the exploration safety but also increase learning efficiency. V-REP and the SL framework can be used to simplify the simulation environment construction.

3.3 Research Problem

Can robots autonomously and safely learn manipulation skills from vision-based observation and exploration in the real world?

Specifically, there are 3 sub-questions as follows:

1. How can robots understand manipulation tasks from observing what they see, including recorded demonstration videos and real-time demonstrations?

2. How can robots learn new manipulation skills and improve mastered skills through exploration? And how to guarantee the effectiveness of the learned skills in the real world?

3. How can the safety of the learning process be guaranteed, particularly in the exploration?
4 PROGRAM AND DESIGN OF THE RESEARCH INVESTIGATION

4.1 Objectives, Methodology and Research Plan

The major objectives of this research are:

1. Develop effective learning algorithms to enable robots to understand manipulation tasks from observing demonstrations.

2. Develop feasible and efficient algorithms to enable robots to master manipulation skills through trial and error.

3. Construct a safe exploration mechanism to reduce the risk in the learning process to as low level as possible, and develop risky behaviour elimination methods as complements to guarantee an absolute safety.

All the algorithms and software will be developed based on ROS, a robot operation system widely used in the Robotics community. Experiments and tests will be made on a general arm robot platform such as Baxter, a general purpose dual-arm robot platform with visual feedback.

4.1.1 Visual Perception Method

The monocular vision with the input form of raw pixel information worked well in playing 2-dimensional (2D) Atari 2600 games [18], but its effectiveness is doubtable in 3-dimensional (3D) environments, because of its impossibility to get depth information. In contrast, the stereo visual perception has the possibility to get depth information. To determine which visual perception method to adopt, comparisons will be made in vision types (e.g., binocular and monocular vision), input forms (e.g., raw pixel information, super pixel, histogram, and semantic map), and color spaces (e.g., RGB, YUV and HSV). The comparison will focus on their influences to the performance of learning algorithms.

4.1.2 Learning from Observation

The algorithm of learning from observation is mainly used to get reward functions to assess various manipulation tasks, i.e., understand manipulation tasks and represent the learned knowledge in the form of reward functions. The reward functions will be used in the implementation of learning from exploration (RL). The algorithm will first be developed based on the method of learning cooking action plans from Youtube videos [38]. For further improvements, other than increasing its learning feasibility and efficiency, research will be conducted to make learning from observation autonomous through unsupervised learning methods.

4.1.3 Learning from Exploration

The algorithm of learning from exploration is applied to master manipulation skills through trial and error. It will be preliminarily developed based on DQN [15]. The preliminary algorithm will be tested through a simple object-picking task. In further research, a post-perception unit will be constructed to select actions for complex manipulation tasks, using a DNN or some
other feasible methods. In the post-perception unit, the safety of the selected actions will also be taken into account. In addition, to develop an algorithm more suitable for real robotic applications, some other kinds of RL algorithms will be developed and compared with DQN according to their learning effectiveness and efficiency.

4.1.4 Safety Guarantee

The safety in the learning process will be guaranteed by a safe exploration mechanism and a risky behaviour elimination unit. As discussed in Section 3.2.5, to keep a good balance between safety and effectiveness, the exploration will be first conducted in simulation environments then fine-tuned in the real world. Simulation environments will be constructed based on V-REP [23]. The feasible methods to reduce the differences between simulation environments and the real world will be studied.

The risky behaviour elimination will be preliminarily implemented through constructing a subset of safe policies by humans [20]. More feasible and efficient risky behaviour elimination methods will be studied in further research. In addition, effective transfer learning methods will be developed to transfer manipulation skills learned in simulation environments to the real world. The transfer learning methods will also be used to transfer mastered skills to new situations.

4.2 Resources and Funding Required

General arm robot platforms and perception sensors, e.g., Baxter robot, NAO robot, MICO robotic arm, cameras, and 3D sensors, which are already in the ARC Centre of Excellence for Robotic Vision at QUT node.

4.3 Individual Contribution to the Research Team

I will carry on this research project under the supervision of Professor Peter Corke, Dr Jürgen Leitner, Associate Professor Ben Upcroft, and possibly some external supervisors. I will be responsible for the design, implementation, evaluation, and results publication of this research project.
### 4.4 Timeline for Completion

<table>
<thead>
<tr>
<th>Time Elapsed (in months for 3 yr study)</th>
<th>3</th>
<th>6</th>
<th>9</th>
<th>12</th>
<th>15</th>
<th>18</th>
<th>21</th>
<th>24</th>
<th>27</th>
<th>30</th>
<th>33</th>
<th>36</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PhD Milestones</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stage 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confirmation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Annual Progress</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final Seminar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thesis Submission</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Coursework</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INN700 Introduction to Research</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IFN001 Advanced Information Research Skills</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Research Process</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Literature Review</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Framework</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preliminary Experiments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learning from Observation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learning from Exploration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Safety Guarantee Method</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Improvement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze Results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Thesis Writing (by chapter)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Title &amp; Abstract</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Introduction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Literature Review</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Framework</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learning from Observation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learning from Exploration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exploration Safety</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Results and Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discussion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conclusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Outputs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conference Papers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Journals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 5 REFERENCE LIST


### 6 Appendix

#### 6.1 Research Ethics Integrity and Safety (REIS)

I have completed and attached the completion certificate for both REIS Quizzes as an appendix to my Stage 2 Report in the form of independent documents.

#### 6.2 Coursework

- INN700 Introduction to Research
- IFN001 Advanced Information Research Skills (AIRS)
6.3 Ethical Clearance Statement

The research will not involve humans, animals, genetically modified organisms or biosafety considerations. It will not involve questionnaires or surveys. No expectation of interacting with people or living species in the course of research.

I have discussed ethics clearance with my supervisory team and we have determined that it is not required.

6.4 Intellectual Property Statement

I need but have not yet signed an IP Assignment Agreement. I will sign an IP later.

6.5 Health and Safety Statement

This research does not involve the use of high risk materials, but a general purpose robot platform. Regarding the robot platform, I am still discussing Health & Safety training with my supervisory team.

6.6 Collaborative Arrangement Statement

I do not require a Collaborative Agreement.

6.7 Memorandum of Understanding for External Supervisors Statement

I do not require a Memo of Understanding as I have no external supervisors at this stage, but I might find some external supervisors in the future.

6.8 Data Management Statement

I have not yet considered data management issues but will do so before confirmation.