

A Probabilistic Path Planning Solution for Industrial Robot
in Dynamic Environment using A Simultaneous Localization
and Mapping Solution

BY

HAFIZ BIN IMAN

A thesis submitted in fulfillment of the requirement for the
degree of Doctor of Philosophy in Mechatronics Engineering

Kulliyah of Engineering
International Islamic University Malaysia

OCTOBER 2022

ABSTRACT

An industrial robot is isolated into work cells making the operation of industrial robots rigid and inflexible. The robot requires tremendous amount of time and resources should a new task or change is required in the work-cell. This stigma and the reality of owning an industrial robot hinder the confidence of Small and Medium-Sized Enterprise (SME) to adopt industrial robot technology. This thesis attempts to democratize robot technology and automation to the SMEs by introducing an inexpensive, flexible, and safe robotic technology. This thesis intends to show the feasibility of constructing a robot manipulator with obstacle avoidance motion planning. Therefore, a six-axis manipulator as a prototype is developed to investigate the planning and motion control of the manipulator in a dynamic environment. The prototype replicates the typical 6R industrial robot design. Rapidly-Exploring Random Tree (RRT) is chosen to plan and control the motion of the manipulator in the dynamic environment. A physic engine simulator simulates the manipulator and the obstacle avoidance planning. A three-dimensional (3D) obstacle is introduced into the simulation environment and moves periodically from an initial point to a goal point in the form of sinusoidal motion described by $A + B \sin(2\pi t C)$. The motion planning is validated with the prototype hardware. However, the moving obstacle is augmented by the simulation environment to maintain safe experimentation and the preservation of the prototype. It is observed that the global planner, RRT, behaves reactively when subjected to the cycle space defined in this report. Consequently, Real-Time Appearance-Based Mapping (RTAB-Map) is used to facilitate an encoder-less context of the manipulator. Although RTAB-Map provides the confident state estimation of the robot task space and, consequently, the joint-space configuration of the manipulator, the estimation is intermittent. It is recommended that a new pipeline, capable of splining between the state estimation and the planning trajectories, is implemented for motion planning in the dynamic environment using Simultaneous Localization and Mapping (SLAM) solutions under the encoderless context.

اضة ضارة

في خلايا العمل جعل تشغيل الروبوتات الصناعية جامداً وغير مرن. الانسان الالي يتطلب قدرأً هائلاً من الوقت والموارد في حالة الحاجة إلى مهمة جديدة أو تغيير خلية العمل. هذه الوصمة وحقيقة امتلاك روبوت صناعي يعيقان ثقة المؤسسات الصغيرة والمتوسطة الحجم (ضش زس) لاعتماد تكنولوجيا الروبوت الصناعي. تحاول هذه الأطروحة إضفاء الطابع الديمقراطي على الروبوت التكنولوجي والأتمتة للشركات الصغيرة والمتوسطة من خلال تقديم وسيلة غير مكلفة ومرنة وآمنة للتكنولوجيا الروبوتية. تهدف هذه الأطروحة إلى إظهار جدوى بناء مناور روبوت مع تخطيط حركة تجنب العوائق. وبالتالي، ستة محاور (٦ض) تم تطوير مناور كنموذج أولى للتحقيق في التخطيط والتحكم في الحركة للمعالج في بيئة ديناميكية. يكرر النموذج الأولى تصميم الروبوت الصناعي النموذجي ٦ض. يتم اختيار شجرة عشوائية سريعة الاستكشاف (ضضة) للتخطيط والتحكم حركة المتلاعب في البيئة الديناميكية. محاكى محرك فيزيائى يحاكي المناور و تخطيط تجنب عقبة. يتم إدخال عقبة ثلاثية الأبعاد في بيئة المحاكاة ويتحرك بشكل دوري من أول أشر إلى نقطة هدف في شكل الحركة الجيبية الموصوفة $A + B \sin(2\pi t C)$. تخطيط الحركة تم التحقق من صحتها باستخدام جهاز النموذج الأولى. ومع ذلك ، فإن التحرك يتم زيادة العائق من خلال بيئة المحاكاة للحفاظ على التجريب الآمن والحفاظ على النموذج الأولى. لوحظ أن يتصرف المخطط العالمي ، ضضة ، بشكل تفاعلي عند تعرضه لمساحة الدورة المحددة في هذا التقرير. وبالتالي ، يتم استخدام ضةا-ش لتسهيل سياق أقل من المشفر المتلاعب. على الرغم من أن ضةا-ش توفر تقدير الحالة الواقع مساحة عمل الروبوت ، وبالتالي تكوين الفضاء المشترك المتلاعب ، التقدير متقطع. من المستحسن أن يكون خط أنابيب جديد ، قادر على الانقسام بين تقدير الحالة ومسارات التخطيط ، هو تم تنفيذها لتخطيط الحركة في البيئة الديناميكية باستخدام حلول ضش اش في سياق غير مشفر.

APPROVAL PAGE

The thesis of Hafiz Bin Iman has been approved by the following

.....

Md. Raisuddin Khan
Supervisor

.....

Md. Mozasser Rahman
Co-supervisor

.....

Norsinnira Bt. Zainul Azlan
Co-supervisor

.....

Hasan Firdaus Bin Mohd Zaki
Co-supervisor

DECLARATION

I hereby declare that this thesis is the result of my own investigation, except where otherwise state. I also declare that it has not been previously or concurrently submitted as a whole for any other degrees at IIUM or other institutions.

Hafiz Bin Iman

Signature

Date

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

DECLARATION OF COPYRIGHT AND AFFIRMATION OF FAIR USE OF UNPUBLISHED RESEARCH

A PROBABILISTIC PATH PLANNING SOLUTION FOR INDUSTRIAL ROBOT IN DYNAMIC ENVIRONMENT USING A SIMULTANEOUS LOCALIZATION AND MAPPING SOLUTION

I declare that the copyright holder of this thesis are jointly owned by the student and IIUM.

Copyright ©2022 Hafiz Bin Iman and International Islamic University Malaysia. All rights reserved.

No part of this unpublished research may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without prior written permission of the copyright holder except as provided below

1. Any material contained in or derived from this unpublished research may only be used by others in their writing with due acknowledgement.
2. IIUM or its library will have the right to make and transmit copies (print or electronic) for institutional and academic purpose.
3. The IIUM library will have the right to make, store in a retrieval system and supply copies of this unpublished research if requested by other universities and research libraries.

By signing this form, I acknowledged that I have read and understand the IIUM Intellectual Property Right and Commercialization policy.

Affirmed by Hafiz Bin Iman

.....

Signature

.....

Date

... Tesis tok saya tuju kepada bapak, mak, Dedek, dan 'ninggal wa rompuan ...

ACKNOWLEDGEMENTS

All glory to the Lord, Allah the Almighty. This research has taught me so much. I have been at my lowest and highest within this endeavor, and of all people that help me with my bearing, none so important as my supervisor, Prof. Dr. Md Raisuddin Khan for taking a chance on me and my concept, and for supporting me endlessly throughout of this research. I thank you and appreciate you and your trust. I hope someday, I could do the same to another brilliant, yet so broken, scholar who only need just one person to help him/her/they shine so bright.

I would be remissed if I did not mention my beautiful family; my mother, Jimelah, my father, Iman and my brother, Fariz. Thank you for being so patient throughout the time I attempt to make this research a successful one. You have been my rock, and will always be my home.

Richard Herman Wallis, the hero of my life and the backbone of this thesis and myself; I cannot thank you enough for always being there to pick me up when I had fell so low.

To my dearest and closest friends, Kirunnisa and Akbar, thank you for the long chats and for being my moral compass. You both are the most beautiful person that I have ever met, inside and out.

To my vai from another mother, (Prof.) Nayeem-vai, thank you so much for being present this whole time when I needed your opinion, your wake-up calls, and your sense of humor. I hope I get to see you again and make robots together this time!

I would also like to thank everyone in the Department of Mechatronics Engineering for being the best example of professionalism. Thank you Dr. Ali Sophian, for having my back and for your vote of confidence.

I also would like to express my gratitude to Ankit-sensei, and Emaru-sensei who brought me in and showed how a world-class institution works and contributes to research and science.

TABLE OF CONTENTS

Abstract	ii
Abstract in Arabic	iii
Approval Page	iv
Declaration	v
Copyright	vi
Acknowledgements	viii
List of Tables	xi
List of Figures	xii
List of Abbreviations	xiv
List of Symbols and Notations	xv
 CHAPTER 1: Introduction	 1
1.1 The Motivation of Eye-in-Hand Robot Configuration	3
1.2 The Devoid of Unified Solution for Uncertainty Management in State and Workspace of an Industrial Robot	4
1.3 Problem Statement and its Significance	7
1.4 Research Philosophy	8
1.5 Objectives	8
1.6 Research Scope	9
1.7 Methodology	9
1.8 The Outline of the Thesis	11
 CHAPTER 2: Literature Review	 13
2.1 Introduction	13
2.2 Uncertainty Management Through State Estimation	14
2.2.1 Extended Kalman Filter for Manipulator's State Estimation	15
2.2.2 Unscented Kalman Filter for Manipulator's State Estimation	17
2.2.3 Particle Filter for Manipulator's State Estimation	18
2.2.4 Miscellaneous Estimation Problem in Manipulator Robot	19
2.3 Map-Building Approach to Uncertainty Management	22
2.3.1 Point Clouds Mapping	22
2.3.2 Occupancy Grid Map: The Map of Uncertain Environment	24
2.3.3 Octomap: The definitive Uncertainty Management Dynamic Environment	26
2.3.4 Miscellaneous Mapping Techniques	27
2.4 Simultaneous Localization and Map-Building as a Total Solution to Uncertainty in a Manipulator's State and Environment	31
2.5 The Planning for 6R Robots in Dynamic Environment	35
2.5.1 The Probabilistic Motion Planning	35
2.6 Summary	37
 CHAPTER 3: Mathematical Backgrounds	 38

3.1	The Kinematics of Richard Mini	38
3.1.1	The Mechanical Descriptions	38
3.1.2	Forward Kinematics of r_{mini}	40
3.1.3	Inverse Kinematics of r_{mini}	43
3.2	r_{mini} 's Path Planner	43
3.2.1	The Cycle Space	46
3.3	The Spherical Harmonics	47
3.4	The Use of Kinematics Jacobian and the Singularities of the Manipulator to Relax the Cost Function	49
3.5	SLAM for r_{mini}	50
3.6	Summary	53
	CHAPTER 4: Experimentation, Result and Discussion	54
4.1	Benchmarking Experiment Design and Result on the Sampling-Based Planner in Static Environment	54
4.2	Experiment Design for Unpredictable Obstacles using RRT	55
4.3	Experiment Design for Moving Planning in Dynamic Environment	60
4.4	Results on Planning for Static and Dynamic Obstacle	60
4.4.1	Result on Path Planning for Unpredictable Static Environment	61
4.4.2	Result on Path Planning Under Dynamic Environment	62
4.4.3	Result on Obstacle-less Planning with SLAM	66
4.5	Summary	69
	CHAPTER 5: Conclusion And Recommendation	70
5.1	Conclusion	70
5.2	Recommendation for Future Works	71
	REFERENCES	71

LIST OF TABLES

Table 3.1	DH-parameter table	42
Table 4.1	The simulated and hardware-connected result of the performance of RRT in a dynamic space. NC stands for No Collision after 5- minute runtime	62

LIST OF FIGURES

Figure 1.1	Three aspects of an FAS to maintain a safe and cost effective robotic system in a production line.	2
Figure 1.2	The second aspect of an FAS consist of two mathematical model that manage the uncertainty of the workspace of a robot and the uncertainty of the state of the robot.	3
Figure 1.3	Eye-in-hand configuration that uses visual feedback enables an articulated robotic arm to identify objects in its workspace for manipulation.	4
Figure 1.4	Eye-in-hand configuration where an RGB-D sensor (blue) is placed on the end-effector link to perform the robot's state estimation via auto-calibration (Klingensmith et al., 2016)	5
Figure 1.5	The gap (labeled as a question mark "?") of a solution to path-planning for uncertainty in the state of the robot and the uncertainty of the environment of the robot in a single framework hinder a functioning FAS in context of an industrial robot	6
Figure 1.6	The variables parameterizing the SLAM solution	7
Figure 1.7	The connection between an FAS to a SLAM solution	8
Figure 1.8	The scope of this research and its considerations	10
Figure 1.9	Summary of the methodology to achieve the objectives of this research	12
Figure 3.1	<i>r_mini</i> wrist conforms to Pieper condition where axis of rotation for joint4, joint5, and joint6 share points of intercept. The dash circle in the diagram a possible point of intercepts. Both point are valid for a Pieper condition	39
Figure 3.2	The location and orientation of <i>r_mini</i> . The choice of the orientation for each frames are based on Denavit-Hartenberg. The joints are values represented by the angle between two x – axes around the z – axis or rotation axis of each actuators	40
Figure 3.3	<i>r_mini</i> and its links	41
Figure 4.1	The top view of the simulation shown in ,(a) , and the isometric view of the benchmark setup in (b). In (c) <i>r_mini</i> attempts to move around the static obstacle placed in it's immediate configuration workspace.	56
Figure 4.2	The benchmark result when two configurations are defined and pass to the OMPL planner pipeline. All planners completed a 50-cycle query from an initial pose to a goal pose. RRT requires the least amount of processing time at finding the motion planning solution, followed by the PRM.	57

Figure 4.3	The experimentation setup for unpredictable static obstacles where the initial and the goal poses are defined in <i>equation 4.2</i> . The green marker shows the initial pose and the red marker shows the goal pose.	58
Figure 4.4	The Gazebo Kinect simulation scene is shown in the <i>gazebo_simulation :: /kinect/rgb/image_color/compressed</i> window and the feedback representation with octomap map model are shown in the backdrop of the figure. Voxelization coloring is based on the height value. Green voxel clustering shows the representation of the static obstacle in the planning scene. In this diagram, the manipulator is in the fifth cycle space.	59
Figure 4.5	The angular displacement and angular velocity for joint1, joint2, joint3, are shown with the planner process time represented by the stick plot. The inset pictures shows the snapshot of the experimentation. The timestamp are represented by the Unix Epoch time format.	61
Figure 4.6	This sequence shows the manipulator follows an outdated trajectory and collides with the cylinder despite attempt to move away from the moving cyclinder.	63
Figure 4.7	The chronology of attempts at avoiding a moving obstacle when the obstacle approaches the robot. The planning algorithm fails at avoiding the cylinder before it passes the $x - location$ of the poses, $c_{initial}$ and c_{goal} . (c) shows the planner successfully provide a non-colliding solution when the cylinder is moving away from robot. (d) shows the Gazebo as the physic engine to replicate the robot hardware and encoders feedback and the cyclical space initialization.	64
Figure 4.8	The sequence of motion when r_{mini} successfully avoid a moving obstacle when the obstacles at a turning point to move away from the hardware.	65
Figure 4.9	Reaction from joint1, joint2, and joint3 shows that the planner together with the cycle space behave reactively towards the moving object. No rapid movement or rate on the last three joints on r_{mini}	66
Figure 4.10	The odometry estimate of the end-effector's frame visualize for the \mathbb{R}^3 part of the task space, C_{ee}	67
Figure 4.11	The odometry and RTAB-Map state estimation output compared together.	68
Figure 4.12	This screenshot shows the visual output of the RTAB-Map, where the state estimation and also the visual odometry together with the map of the environment are represented on the left pane of the snapshot. The right pane shows the initial and the goal pose for the C_{cycle}	68

LIST OF ABBREVIATIONS

3D	three-dimensional
AIEKF	Adaptive Iterative Extended Kalman Filter
ARA*	Anytime Repairing A*
ARM-SLAM	Articulated Robot Motion for Simultaneous Localization and Mapping
AXBAM	Autonomous eXploration to Build A Map
BNM	Best Next Move
CPF	Conventional Particle Filter
DOF	degree-of-freedom
EKF	Extended Kalman Filter
EKF-RLFJ	Extended Kalman Filter and Rigid Link Flexible Joint coupling
FAS	Flexible Automation System
GAF	Group Average Feature
GMM	Gaussian Mixture Model
GMR	Gaussian Mixture Regression
HAMP-U	Hierarchical and Adaptive Mobile Manipulator Planner Uncertainty
ICP	Iterative Closes Point
IDM	Inverse Dynamic Model
IMU	Inertial Measurement Unit
IPA	Infrared Proximity Array
LiDAR	Laser Imaging, Detection and Ranging
LM	Levenberg-Marquardt
LSD-SLAM	Large-scale Direct mono-SLAM
MPF	Manifold Particle Filter
PHASER	Phase Spherical Harmonics approach to map registration
PRM	The Probabilistic Roadmap
r_mini	Richard Mini, a compliant six-axis manipulator
RAMP	Real-time Adaptive Motion Planning
RANSAC	Random Consensus
RGB-D	Red-Green-Blue-Depth sensor or visual-depth camera/sensor
RLFJ	Rigid Link Flexible Joint
RRT	Rapidly-Exploring Random Tree
RTAB-Map	Real-Time Appearance-Based Mapping
SLAM	Simultaneous Localization and Mapping
SME	Small and Medium-Sized Enterprise
srUKF	Square Root Unscented Kalman Filter
SSPF	Scaling Series Particle Filter
SURF	Speed Up Robust Feature
TSDF	Truncated Signed Distance Field
UKF	Unscented Kalman Filter
UWB	Ultra-wideband

LIST OF SYMBOLS AND NOTATIONS

A_i	Homogenous transformation of rigid body in 3D
C_{cycle}	Subset of the configuration space defined by the cycle space generator, algorithm 3
C_{ee}	Set of configuration space for the end-effector on $\mathbb{R}^3 \times SO(3)$
C_{free}	Non-colliding configuration space
C_{limit}	Constrained imposed on the configuration space
C_n	Set of configuration space on 3D, homeomorphic to the open set of constricted 6-sphere space
C^n	Control space, specifically the n-joint robot manipulator
$C_{obstacle}$	Occupied space, i.e. obstacles in the worksapce and the robot arm itself
\tilde{C}	Correlation on 2-sphere
$C_{workspace}$	Valid reachable pose constrained by the joint angles
D	The Weigner-D matrix or function
\mathbb{H}	The Hamiltonian space
J	The kinematic Jacobian
$L(n)$	The logit function
\mathcal{M}	Collision map in m_i
P_m^l	The Legendre Polynomial
$P(n)$	Posterior probability function for the occupancy of a voxel
\mathbb{R}^3	3D space specifically refering to the Euclidean space
$Rot_{axis,angle}$	A 4×4 matrix encoded with R rotation along an axis, at an angle
R	Rotation matrix
S^2	The topology of a 2-sphere
$SE(3)$	Special Euclidean space in 3D space
$SO(3)$	Special orthogonal group in 3D space representing rotation and rotational algebra
$T_{axis,angle}$	A 4×4 matrix encoded with t translation along an axis, at an angle
\mathcal{T}	Tree node structure in m_i
Y_m^l	The spherical harmonics component of the spatial Fourier transform
a_i	Denavit-Hartenberg's z – axis offset parameter
$Z - Y - Z$	The Z-Y-Z Euler angles
$x - \text{axis}$	The x-component of the Cartesian coordinate system
$y - \text{axis}$	The y-component of the Cartesian coordinate system
$z - \text{axis}$	The z axis on the Cartesian coordinate system
c_{ee}	Element in configuration space at the end-effector or the task space
c^{ee}	Element in control space at the end-effector or the task space
\hat{c}^{ee}	Estimated element in control space at the end-effector or the task space
c_n	Element in the configuration space, C_n
c^n	Element in the control space, C^n
c_{random}	Random sampling in the configuration space
d_i	Denavit-Hartenberg's x – axis offset parameter

\tilde{f}	A target 3D pointcloud treated as a function
\tilde{g}	A 3D pointcloud treated as a function
m_i	Global or local map
n	Node representing the discretization of 3D space called voxel
p	Bayes' posterior
q	The 4-vector parameterization of the quaternions in the Hamiltonian space
t	Time
t	Translation 3-vector
u_i	State transition model
\hat{x}_i	State estimation in 3D space
z_i	Observation model
\boxplus	A Gaussian fusion technique
Δt	Distance between two points in 3D
Φ	Map represented by the Octomap-harmonics posterior from the spherical harmonics state estimation pipeline
ψ	The latitude of the 2-sphere
τ	Trajectory in control space, C^n
θ	The longitude of the 2-sphere

CHAPTER 1

INTRODUCTION

The background of this research centers around shifting the usage of industrial robot from large enterprises to the small and medium size business. This thesis loosely refers an industrial robot as a robot manipulator that is used in automation per definition ISO-8373:2 (2021). Thus, any manipulators with more than three controllable joints used in an automation for production purposes are considered as an industrial robot. However, the stigma prevails; industrial robots are heavy, expensive, inflexible, high maintenance, and hazardous which requires informed safety precautionaries. In practice, a heavy industrial robot is isolated into workcells making the operation of industrial robots rigid, inflexible, and requires tremendous amount of time and resources should a new task or change is introduced in the workcell (Miseikis et al., 2017). This stigma and the reality of owning an industrial robot hinders the confidence of SME to adopt industrial robot technology. This thesis attempts to democratize robot technology and automation to the SMEs by introducing an inexpensive, flexible, and safe robotic technology.

I propose a Flexible Automation System (FAS) to increase flexibility and decrease the cost of operating and maintaining an industrial robot. The FAS is characterized by its ability to react to unpredictable changes in its production floor using a SLAM solution so that my FAS solution can be installed in a large-volumed production floor under an SME setup. The main purpose of the FAS-SLAM system is to maintain and manage the uncertainty of the system so that the system will be safe to use at a low cost. In the coming sections I will establish the connection between an FAS system and a SLAM solution. As a primer, there are three aspects of an FAS for an industrial robot;

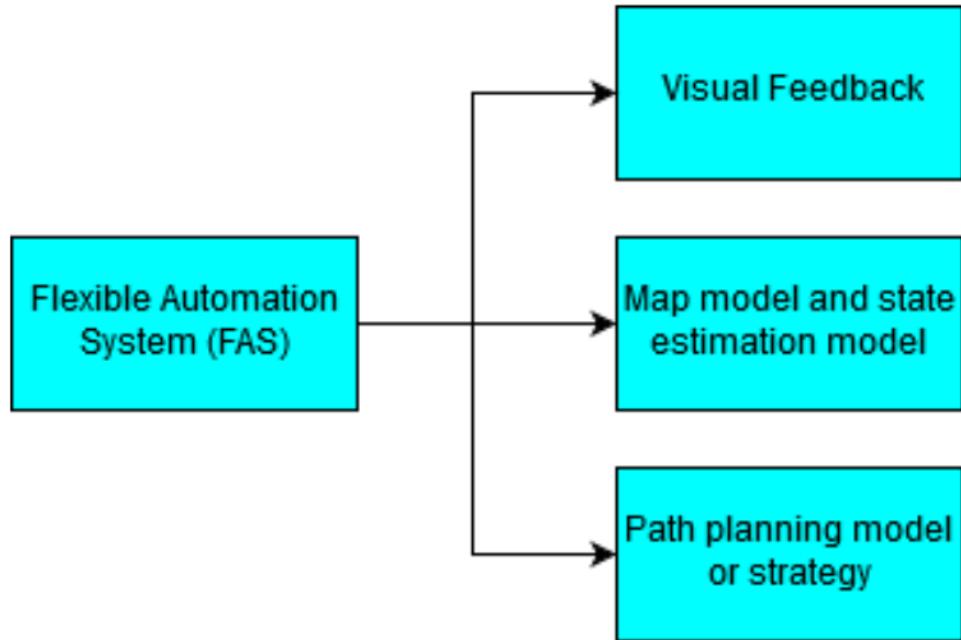


Figure 1.1: Three aspects of an FAS to maintain a safe and cost effective robotic system in a production line.

(1) visual feedback, (2) map model and state estimation model of the robot, and (3) path planning model of the robot. Figure 1.1 shows these considerations.

The FAS uses visual feedback, such as visual camera, laser range finder, or a visual-depth camera, to model the workspace and to model the state of the robot.

The second aspect of an FAS is the state estimation model and the map model summarized by figure 1.2. A map model is a mathematical representation of an environment and state estimation model is a process of estimating an industrial robot configuration, location, velocity and acceleration of its end effector. The map model will provide the information for the FAS to manage the movement of a robot arm in its workspace based on the state estimation.

The third aspect of an FAS is the path-planning model. The path-planning model is used to calculate a way to reach a point in space without colliding with any obstructions or obstacles. The path is dependent on the information restored in the map model of the workspace.

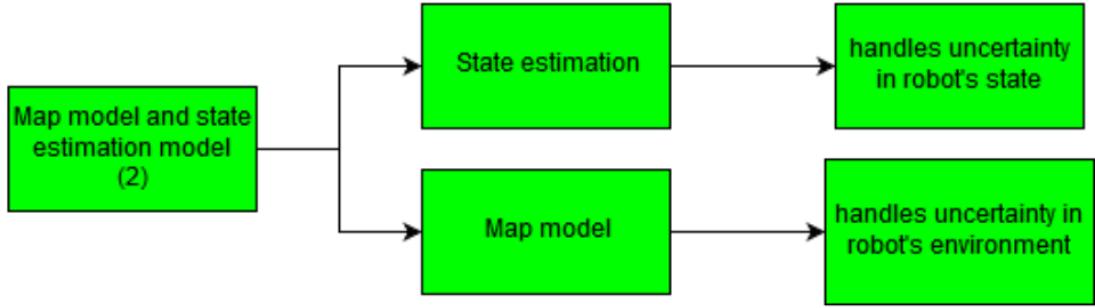


Figure 1.2: The second aspect of an FAS consist of two mathematical model that manage the uncertainty of the workspace of a robot and the uncertainty of the state of the robot.

1.1 THE MOTIVATION OF EYE-IN-HAND ROBOT CONFIGURATION

Placement of the machine vision and the decision of the placement of the vision is non-trivial. The designer can consider, eye-to-hand configuration where the vision sensor is attached to an additional structure that has a vantage point of the robot manipulator and the workspace (Luo & Kuo, 2016) or an eye-in-hand configuration where a vision sensor is place on the robot's end-effector or at the last link of the robot arm. The latter configuration requires no additional structure and the visual feedback can be used as a state estimator and a mapping tool abiding the movement of the robot. This makes eye-in-hand configuration more space-efficient. Unlike eye-to-hand configuration, eye-in-hand sensors provide more information gain in terms of the state of the robot and the environment. The feedback from eye-in-hand configuration lacks visual-obstruction where more than one vantage point can be achieve when the sensors move with the end effector. The sensor in eye-in-hand configuration aids task involving reaching and manipulating since both tasks are specific to the end-effector. In the case of eye-to-hand configuration, both reaching and manipulation may be subjected to visual obstruction and extra rigs for the sensor.

As an example, Luo and Kuo (2016) used Microsoft's Kinect, a type of visual-depth sensor Red-Green-Blue-Depth sensor or visual-depth camera/sensor (RGB-D), to produce workspace model of their robot system and to identify objects in the workspace. Figure 1.3 shows the rigidity of their setup.



Figure 1.3: Eye-in-hand configuration that uses visual feedback enables an articulated robotic arm to identify objects in its workspace for manipulation.

Klingensmith et al. (2016) uses the eye-in-hand configuration where an RGB-D sensor is connected at the end effector to auto-calibrate the robot's position and configuration illustrated in figure 1.4. Extra structures were observed in the figure but in this stage of their research, their eye-to-hand sensor setup are not reported in their findings.

1.2 THE DEVOID OF UNIFIED SOLUTION FOR UNCERTAINTY MANAGEMENT IN STATE AND WORKSPACE OF AN INDUSTRIAL ROBOT

Despite rich solution options to uncertainty of a robot state and its environment, the solutions are disjoint and performed separately. SLAM, however, incorporate both the solution to uncertainty of the robot's state and the solution to uncertainty of the environment into one framework. Equation 1.1 summarize the concept of SLAM:

$$p(m_i, x_i | z_i, u_i, x_{i-1}) \quad (1.1)$$

where p (also known as posterior) is the process of maintaining the map of an unknown environment and estimating the current state or pose of a robot. $m_i \in \mathbb{R}^{3n}$, is the global map model, $x_i \in \mathbb{R}^3 \times SO(3)$ is the state estimation, z_i is the measurement or observation model or visual feedback model of the robot, and u_i is the state transition

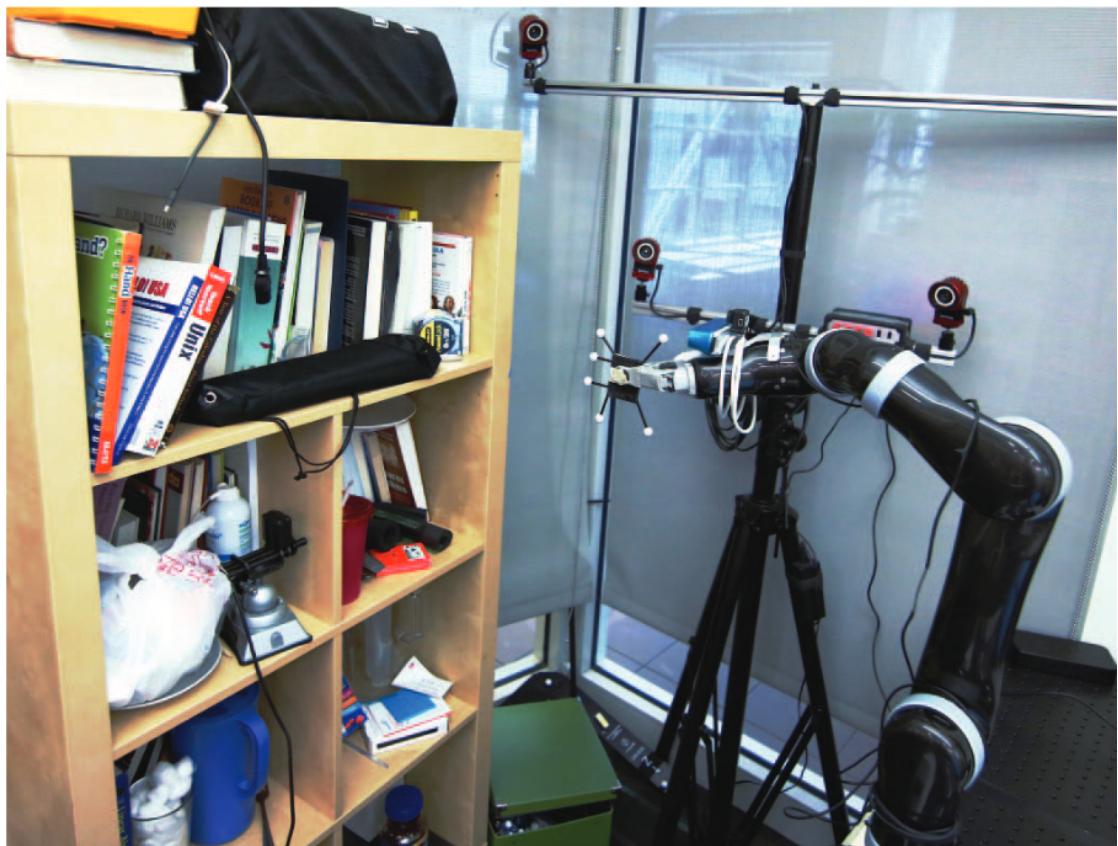


Figure 1.4: Eye-in-hand configuration where an RGB-D sensor (blue) is placed on the end-effector link to perform the robot's state estimation via auto-calibration
(Klingensmith et al., 2016)

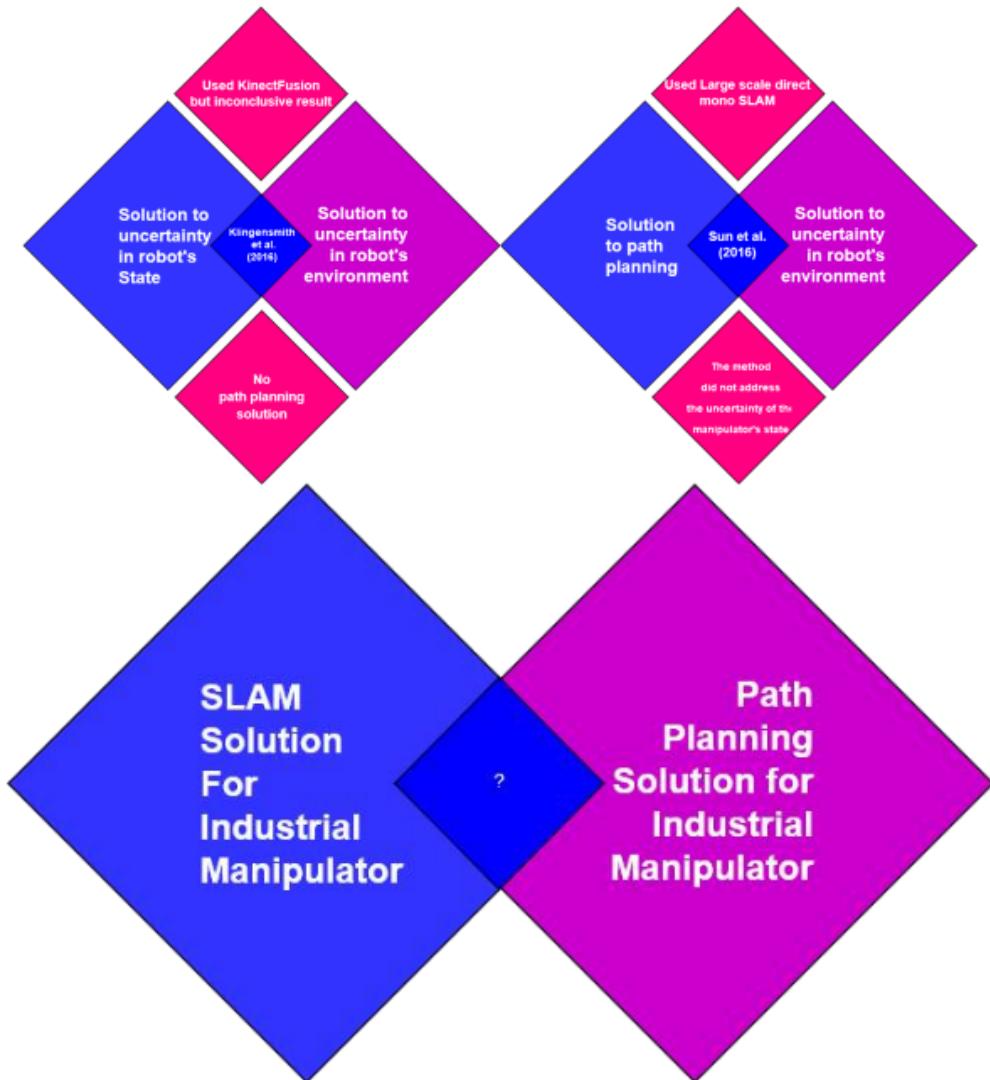


Figure 1.5: The gap (labeled as a question mark "?") of a solution to path-planning for uncertainty in the state of the robot and the uncertainty of the environment of the robot in a single framework hinder a functioning FAS in context of an industrial robot

matrix or state transition model of the robot. x_{i-1} before a new measurement is taken.

Figure 1.6 summarizes the arguments of equation 1.1.

In theory a SLAM solution covers the first and the second aspects of the FAS proposed in this research. Figure 1.7 articulates the relevance of SLAM solution to an FAS.

Yet SLAM has only been optimized specifically for autonomous robot to address an unknown environment. The definitive researches on the use of SLAM in articulated robot were introduced by Klingensmith et al. (2016), Li et al. (2019), and Ito et al.

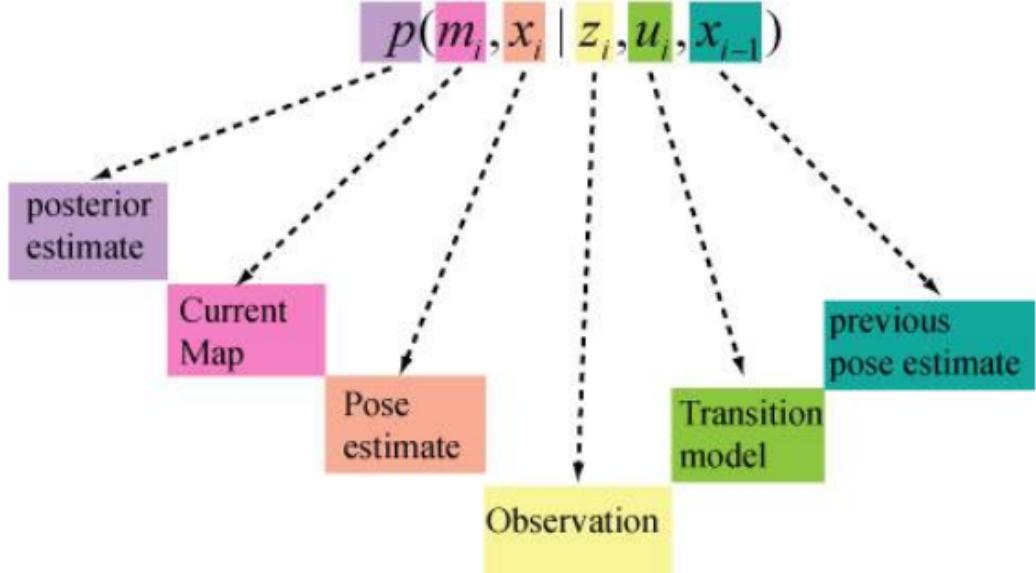


Figure 1.6: The variables parameterizing the SLAM solution

(2020). However, they did not consider the uncertainty of the state of the robot, the uncertainty of the robot's environment, and the path planning solution in a single framework. Figure 1.5 summarizes the gap in finding a solution to a path-planning under the uncertainty of the state of the robot and the uncertainty of its environment.

1.3 PROBLEM STATEMENT AND ITS SIGNIFICANCE

The current state-of-the-art approaches to an industrial articulated manipulator lack a solution that addresses the safety of the system in a changing environment. SLAM solutions for articulated manipulator have only addressed the issues of accuracy without tackling the high maintenance cost and safety of a robot manipulator specifically on the production set-up. Furthermore, the performance of these solutions against probabilistic path-planner for robot manipulator has yet been reported. This research intend to aspire flexibility and cost effective robot manipulator system for industrial purposes in SME's using sampling-based planner closely coupled with a SLAM solution.

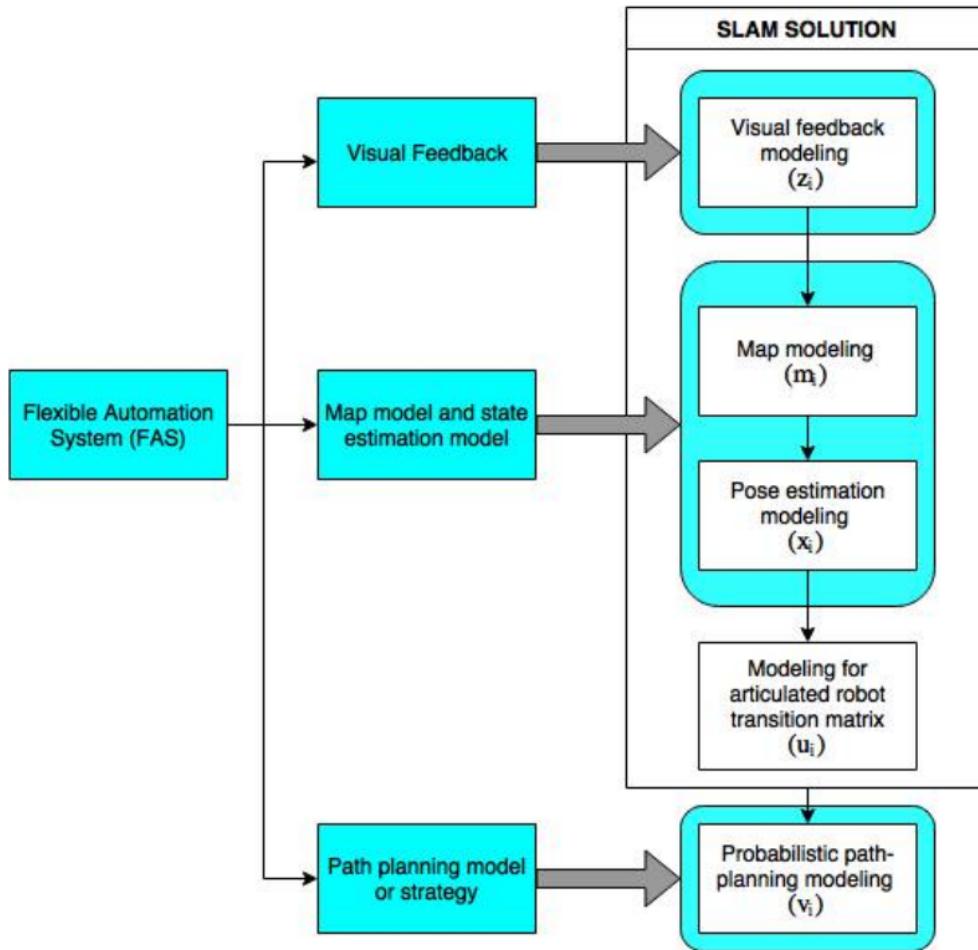


Figure 1.7: The connection between an FAS to a SLAM solution

1.4 RESEARCH PHILOSOPHY

A compliant robotic arm by leveraging the probabilistic mathematical models for map of an environment, the state estimation of a robot, and the path-planning model in controlling the robot motion sustains safety operation and cost-effective production line for SME's.

1.5 OBJECTIVES

1. To design a six-axis manipulator and build it as a prototype of a compliant manipulator.
2. To simulate a moving obstacle avoidance capability using a probabilistic planner.

3. To demonstrate the obstacle avoidance capability on the compliant manipulator hardware with a synthetic moving obstacle augmented from a simulated environment.
4. To show the feasibility of using a SLAM solution as a feedback pipeline in motion planning.

1.6 RESEARCH SCOPE

This research uses a back-drivable (compliant) articulated robot with six axes to implement the framework of a fully probabilistic strategy to path-planning and obstacle avoidance. This research only use an RGB-D sensor.

The dynamic environment is a non-reflective and non-specular workspace. In context of designing the workspace as a dynamic environment, the workspace is not share with another robotic arm. Instead, the workspace will be introduced with a moving obstacle. Figure 1.8 shows the scope and the considerations of this research.

1.7 METHODOLOGY

In this research, the model of a robot kinematics, specifically on the task space (the end-effector frame) of the robot, $C_{ee} \in \mathbb{R}^3 \times SO(3)$ where, $\mathbb{R}^3 \times SO(3)$ is homeomorphic to the special Euclidean group, $SE(3)$. Hence, given $\{C_{ee} = c_{ee}\}$, the task space of the robot manipulator is a set in equation 1.2:

$$\left\{ c_{ee} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} : R \in SO(3), t \in \mathbb{R}^3 \right\} \quad (1.2)$$

where R is the rotation matrix and t is the translation in 3D. Thus, since all SLAM solutions for three-dimensional space provide state estimation in the form of $\mathbb{R}^3 \times SO(3)$, their model in equation 1.1 holds for industrial robot arm. Nonetheless, the complete solution for SLAM does not consider the path-planning model of the robot arm, specifically, the mapping of control space, $C^n \in \mathbb{R}^n$ into the C_{ee} , where n is the number of

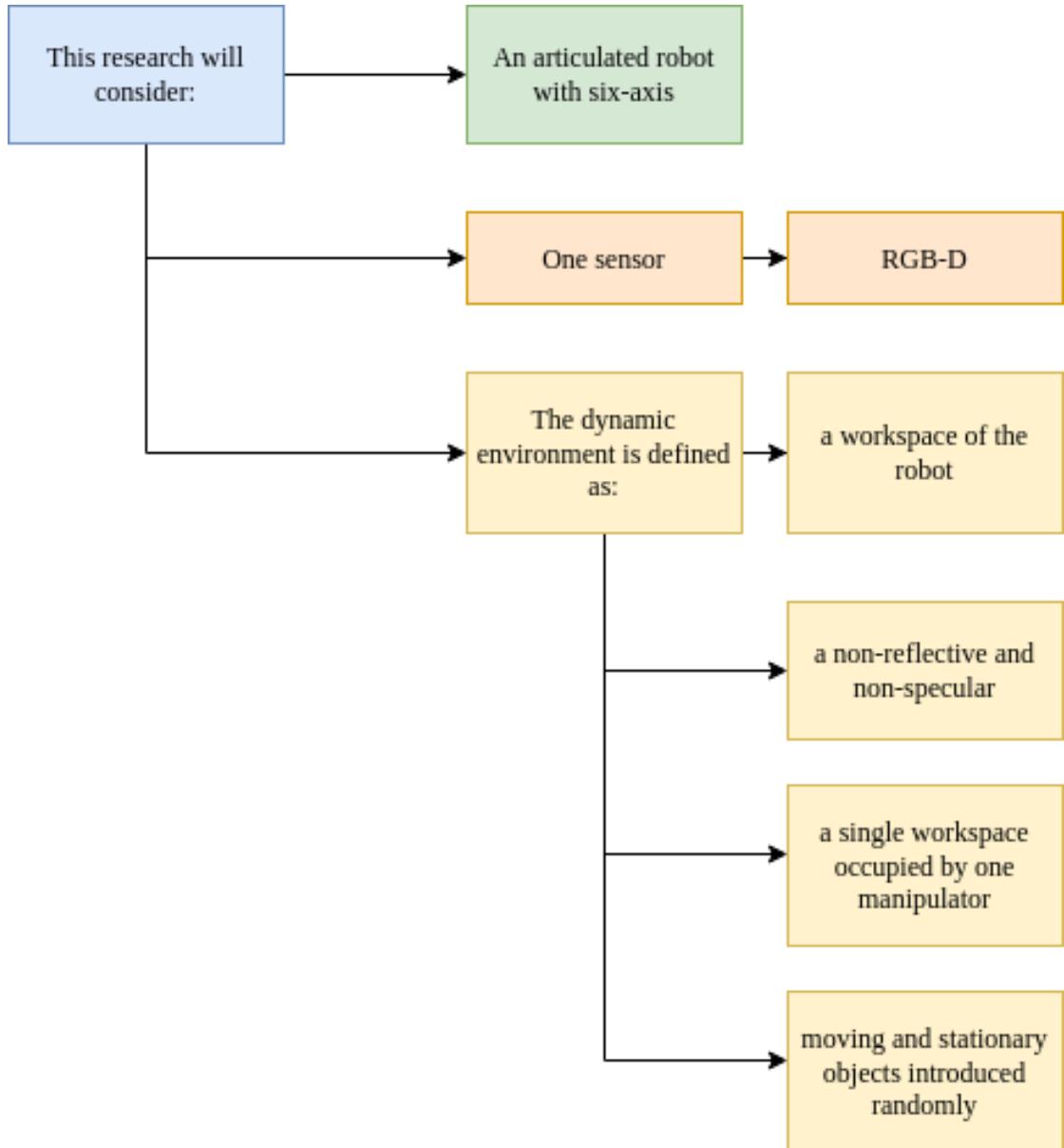


Figure 1.8: The scope of this research and its considerations

rigid body in the robotic arm. Hence, I will investigate the tractability of reconciling probabilistic model of a path-planning strategy with Equation 1.1 such that:

$$p(\mu_i, m | \hat{x}_i, z_i, u_i) \quad (1.3)$$

where p is similar to the process of maintaining the map of the workspace and estimating the state of a robot concurrently where, \hat{x}_i , is the state estimation pipeline of a SLAM solution in equation 1.1.

In equation 1.3, the solution incorporates both SLAM algorithm and a probabilistic path-planning model into a single framework instead of considering the SLAM solution and path-planning algorithm separately. I outline my research methodology based on equation 1.3. An overview of the research methodology against the objectives of this research is presented in figure 1.9.

1.8 THE OUTLINE OF THE THESIS

In this chapter, the concept of FAS is translated into formulating a SLAM solution for a robot manipulator. This thesis will elucidate the SLAM-planner coupling and pipelininig in the coming chapters. The reader is usher to a literature review of the state-of-the-art and the leading papers on state estimation, map-building models and path-planning in chapter 2. The readers are then introduced with the mathematical foundation in chapter 3. In chapter 4, the experimentation are delineated and the chapter continues with the discussion on the findings and result. This thesis concludes with chapter 5 with recommendation on future works.

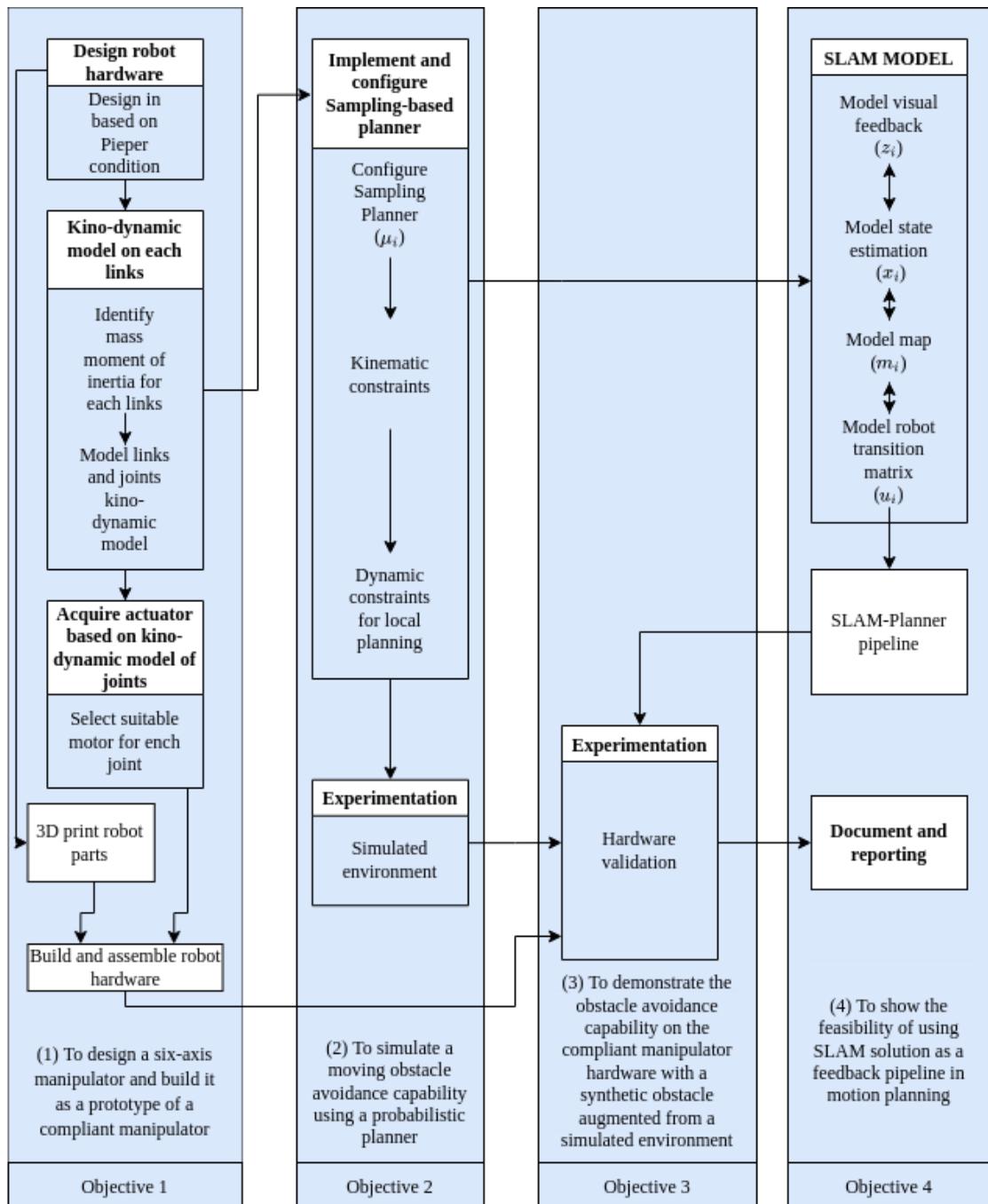


Figure 1.9: Summary of the methodology to achieve the objectives of this research

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, the uncertainties in manipulators and the state-of-the-art approach to model and manage the uncertainty are elaborated and discussed. The following sections use the terms *error* and *uncertainty* interchangeably.

In an industrial environment, the uncertainty of an industrial robot is managed by isolating the manipulator into work cells. It lacks predictive behavior and only function in a predefined setup (Miseikis et al., 2017). Such a system is fragile to unpredictable changes such as an accidental introduction to an object in its environment. The robot system often costs more because of constant reprogramming if predefined changes are required. This review explores solutions that enable the manipulator system to react appropriately to changes in the environment while taking account the uncertainty of its internal state. With these solutions robot manipulator can work in a more efficient setting, i.e., involves less reprogramming and capable of working at close proximity with human operator safely.

This research classifies the uncertainty into two categories; uncertainty in the manipulator's state and uncertainty in the manipulator's environment. The uncertainty in a state is often caused by indirect measurements of the manipulator's joint parameters because of the limitation of sensor choice (Du et al., 2014; Hebert et al., 2012; Janabi-Sharifi & Marey, 2010; Jassemi-Zargani & Necsulescu, 2002; Rigatos, 2009) and the characteristic of the manipulator's joints and links (Biber, 2003; Lertpiriyasuwat et al., 2000; Lightcap & Banks, 2010; Sawada et al., 2012; Ulrich, 2011) The solutions to estimate a manipulator's state involve filtering techniques. The uncertainty in an en-

vironment is often related to a lack of information about the environment's state. The solutions to manage uncertainty in an environment are elaborate and non-trivial.

The following discussion will outline the uncertainty management of a manipulator by using an extended Kalman filter, unscented Kalman filter, and particle filter. The disucssion continues with the uncertainty management of the environment using point clouds and occupancy grid map and its variants. The uncertainty management of the state of a robot and its environment are discussed based on the Simultaneous Localization and Mapping. SLAM solutions are frameworks for more unified uncertainty management in autonomous robot. However, this research will look into their application on robot manipulator.

2.2 UNCERTAINTY MANAGEMENT THROUGH STATE ESTIMATION

Filters are estimation tools used to observe the state of a system. For a manipulator robot, the state of the robot's end effector or its end point is usually confirmed by direct measurement of the joint position. If this direct measurement is unavailable, the state of the robot is measured or estimated using observer or filters.

Filtering techniques uses probabilistic model or framework to handle and manage the uncertainty of observing the state of the manipulator robot. It is often based on Baye's rule where a model of the state is conditioned based on incomplete or indirect measurement. Here, the model of the state is often called a prior and the indirect or incomplete measurement is often regard as the update of a filter. If a prior of a filter is defined by another measurement, we consider the application of the filter as a data fusion technique. After conditioning the measurement update, the state estimation of the robot is ascertained by applying Baye's rule. A filter often involves linearization like the case of an extended Kalman filter. They are also recursive such that the previous estimation is used as the new prior during successive sampling.

In this section, the use of extended Kalman filter, unscented Kalman filter, particle filter and their role at managing uncertainty for data fusion and state estimation are

reviewed.

2.2.1 Extended Kalman Filter for Manipulator's State Estimation

Jassemi-Zargani and Necsulescu (2002) used extended Kalman filter to fuse data from high resolution joint resolver. A joint resolver is a control unit that performs calculation of the inverse transformation of a manipulator from data obtained from the end effector. They use the data from two accelerometers to estimate the position of each joints in their manipulator and use these measurements in their extended kalman filter observer to estimate the acceleration of the end effector that has jerking motion. Their use of two sensors in extended Kalman filter is an example of data fusion technique.

Lertpiriyasuwat et al. (2000) used extended kalman filter to estimate the position of the end effector of two axis robotic arm together with the joint measurement for highly flexible links in real time. The uncertainty in their system is caused by the flexible links. They use a two-link manipulator. Each joint has optical encoder and the end point has a reflective infrared-light emitter. The deflection of each links is modeled using the deflection beam model. The dynamic equations of the manipulator are derived from Kane's method. They linearized the dynamic equation by eliminating the non-linear terms that involve the generalized elastic coordinates and their derivatives in the inertial matrix and the velocity vector. They coined the linearization method as a 'ruthless linearization'. The extended kalman filter is used to estimate the position of the end effector using the differential equation solution of the dynamic equation which was solved using Runge-Kutta method. Their result shows that with ruthlessly linearized model, the extended Kalman filter can estimate the position of the end effector consistently during low-speed and high-speed slew maneuvers compared to continuously linearized model of the dynamic model. From this research, linearization of model before the use of extended Kalman filter affects the performance of the filtering technique.

Ulrich (2011) proposed the use of extended Kalman filter to estimate joint positions and velocities for flexible joint positions and velocities for flexible joint space robotic manipulator. The joint flexibility in manipulators for space robots are obvious

because they are lightweight which introduce uncertainty in estimating joint position. Thus, they extend the design of extended Kalman filters based on nonlinear joint models for use with an adaptive controller. By using this combination, they increase the accuracy of the closed-loop estimation and control of a flexible joint space robot. I observe that the uncertainty of their joints also comes from the linearization of a nonlinear behavior of the flexible joint of their space robots. Although nonlinear joints can be approximated by representing joint flexibility by a linear spring model, the researchers argue that such assumption is inaccurate. Accordingly, they add nonlinear stiffness, soft-windup, frictional losses, inertial cross-coupling to their joint model. They presented a converging error for a non-linear based extended Kalman filter technique used with linear and nonlinear joint model. However, the error diverges for linear based extended Kalman filter techniques with the same model coupling.

Lightcap and Banks (2010) attempted to estimate the configuration of a Rigid Link Flexible Joint (RLFJ) manipulator using discrete-time extended Kalman filter as an observer for the robot model and control system. Because of the flexible joints, direct measurement from the encoders cannot represent the position of the joint directly making the position of the robot's end effector nonlinear. The authors use extended Kalman filter to estimate the pose of the end effector and manage the uncertainty of the flexible joint through linearization. The uncertainties of the link and the motors' dynamics are modeled into the manipulator's dynamic equations. The authors stress that extended Kalman filter has non-optimal estimation as is the case of any algorithm that requires linearization. They performed a simulation and reiterate their models and algorithm on a Mitsubishi PA10-6CE manipulator experimentally. They observed improved tracking performance for highly flexible joints (joints under high torque) and low tracking performance for rigid joints. Thus, the authors introduce a mixed rigid-joint/flexible joint model to an Extended Kalman Filter and Rigid Link Flexible Joint coupling (EKF-RLFJ) controller which improved the overall tracking performance.

During visual servoing of a manipulator, the pose of the end effector is determined from an intial estimate of its position. Janabi-Sharifi and Marey (2010) demon-

strate an Adaptive Iterative Extended Kalman Filter (AIEKF) in the absence of accurate initial pose, noise matrix, and covariance matrix at variant sampling rate during robotic visual servoing. Visual servoing is a process of estimating the configuration of a manipulator using images or visual feedbacks. They experimented on a six degree of freedom cartesian manipulator, AFMA-6, with eye-in-hand camera configuration. The robot moves at a predefined trajectory under different condition. Each separate experimentation involves increasing the velocity of the end effector, changing the covariance matrices, changing the sampling time for estimation, and changing the initial positions of the end effector. They compared three other Kalman filtering techniques (extended Kalman filter, adaptive extended Kalman filter, and iterative extended Kalman filter) with the AIEKF. They conclude that AIEKF can improve pose estimation during uncertain initial position, uncertain covariance matrix estimation, high motion and slow sampling rate.

2.2.2 Unscented Kalman Filter for Manipulator's State Estimation

Haghishipanah et al. (2015) address nonlinearity problem of an elastic cable as power transmission between a motor and a joint for a surgical arm. The coupling between the motor and the robot joint reduce armature mass, inertia and size for expert surgery but the cable elasticity introduces uncertainty and nonlinearity to the kinematics and the dynamics of the surgical arm. They introduce an estimation method that uses an Unscented Kalman Filter (UKF) and a Square Root Unscented Kalman Filter (srUKF) to estimate cable coupling parameter and the position of the end effector (end point) in real-time. The authors implement their method using the Raven-II, a seven degree of freedom serial manipulator, as their surgical arm. The surgical arm is equipped with an optical encoder attached to the motors and position sensor on each joint for data validation. In their research, the authors only address the first three joint for state estimation of the Raven-II. They model the armature dynamics using forward and inverse dynamics with Newton-Euler equations. Their inverse dynamic solution is based on the recursive Newton-Euler algorithm. The Newton-Euler equations requires model parameters. The

authors identify the initial inertial matrices, mass and the center of mass of the system using a computer aided design model and use srUKF to estimate the joint angle and joint position of the surgical arm online. Also, the authors apply the standard UKF to estimate the coupling parameter offline. They compute spring constant, damping constant, coulomb and viscous friction of the motor side and the joint side empirically. They validate their method experimentally using three different design. The first two experiments involve changing the cable tension and the third experiment involve picking an object of the mass 100 g under high cable tension. They compared the result of the three experiment between the dynamic model that uses their UKF estimation and the dynamic model that has no filtering technique. They improved the accuracy of the joints position to 1.4333 respectively. The authors also observe that whe the flexible joint are model as a rigid body, the performance of their unscented Kalman filter deteriorate. They repeat the same problem mix rigid-flexible cable model and replace the encoder with stereo camera (Haghigipanah et al., 2016). They improve the accuracy of the joint position to 43.14%, 33.42%, 72.05% for joints 1, 2, 3 respectively.

2.2.3 Particle Filter for Manipulator's State Estimation

Biber (2003) present the design of two filtering techniques to estimate the acceleration and the jerk of five-bar linkage manipulator with flexible joints. These estimations help in designing a better control. Unlike a rigid joint, the flexible joints are time variant hence highly nonlinear. The nonlinearity introduce uncertainty to the dynamics of the manipulator making prediction of higher order dynamics such as accelerations and jerks uncertain. This is because a nonlinear system has no closed form solution and require linearization. The authors suggest the use of Euler-Langrange equations to represent the model of the manipulator dynamics given that each joint are model by torsional spring. To better estimate the acceleration and jerk of each link in the four-bar linkage, the linearization is manage using extended and unscented Kalman filters. The author validated their observer model by simulation and conclude that the accelaration and jerk of the manipulator's linkages are successfully estimated.

Rigatos (2009) uses particle filter to fuse data from an Inertial Measurement Unit (IMU) and joint encoders to estimating pose of the end effector. The purpose of this paper is to estimate the state vector of a three degree of freedom industrial robot using accelerometer and an encoder for each joint. The estimated state vector is used to generate appropriate control signal for the manipulators. Accelerometer are notorious for being nonlinear. The authors use particle filter to perform data fusion between an accelerometer and an encoder to estimate the pose of the end effector. Readers should note that both accelerometers and encoders has non-gaussian behavior. This behavior is stronger for flexible joints because they introduce nonlinearity to the state (position of the end effector) of the manipulator. Particle filter is used simply because it is nonparametric. This mean that the parameter of a normal distribution are not assumed. Instead, particle filter performs the estimation based on the sampled data and generate the distribution from these samples. The method of sampling provide a general solution that has no presumption of sensor characteristics. This also means that with particle filter, more accessible sensors such as an accelerometer or an IMU can be used without scaling down accuracy. The state vector estimation was compared with extended kalman filtering technique. The authors observe higher accuracy in estimation of the state vector for the particle filtering technique compared to the extended Kalman filter technique. However, the author caution that the selection of particles numbers may improve accuracy with the expense of computational load.

2.2.4 Miscellaneous Estimation Problem in Manipulator Robot

Du et al. (2014) use particle filter to estimate pose for a visual servoing application. The motivation behind this paper is to introduce contactless and markerless control of a manipulator using computer vision. The researcher uses 3D camera (Kinect sensor) to translate a human arm motion into a motion of a manipulator and Camshift program library to track hand position. The particle filter is used to estimate the hand position and orientation. The particle filter handles the noise error from the Kinect sensor and the accumulated error introduced from the Camshift method of tracking. The robot inverse

kinematics are solved numerically using the Levenberg-Marquardt (LM) algorithm.

Du and Zhang (2014) use extended Kalman filter to handle kinematic errors in manipulators. Kinematic errors occur because of imperfection in serial robot components, their wear, misalignment and other factors. The extended Kalman filter allows auto-calibration without strenuous technique and expensive sensors when the kinematic errors are considered. In their method, the IMU and the position sensors are attached at the end point of the six degree of freedom GOOGOL GRB3016 robot. The authors also explain data fusion algorithm for their IMU attached at the end point of their serial robot where a particle filter is used to estimate the orientation of the end point of the serial robot and a Kalman filter is used to estimate the position of the end point of the serial robot. A Kalman filter is the base of both extended and unscented Kalman filter which can only perform well for linear system. The extended Kalman filter is used to optimize the position and orientation estimation of the end point of the robot. By using the Jacobian matrices, the authors estimate the kinematic errors of the serial robot to manage the uncertainty of using IMU measurements. Du et al. (2014) compare their extended Kalman filter approach to a linear least square technique for their estimation of each joints in the robot. The extended Kalman filter has lower error for all six joint parameters estimation. The author acclimate that their method of using IMU and position sensor, via position markers, reduce the complex steps of auto-calibrating a manipulator, increase better accuracy, convenience, and effectiveness.

Hebert et al. (2012) present data fusion algorithm using unscented kalman filter to estimate the manipulator tool and the manipulated object simultaneously. The fusion algorithm is used to manage the uncertainty of the end effector location as a result of uncertain actuation because of unknown weight of the manipulated object. Also, the authors use Barret WAM manipulator that introduces further uncertainty in actuation as a result of tendon actuation similar to flexible cable actuation. These uncertainties prompt the use of two type of sensors as feedbacks, visual and tactile sensing. They use unscented Kalman filter to fuse image features that covers dense range, visual appearance, silhouette of manipulator arm, multi-fingered hand and grasped object. To fuse

these measurement, the authors model three measurements: (1) the measurement model for manipulator's hand tracking by using appearance, shape and silhouette, (2) the object tracking measurement using point cloud association like iterative closest points, and (3) the tactile measurement model to represent a binary state of contact between the fingers of the manipulator and the manipulated object. The DARPA ARM-S with Barret WAM manipulator were used to validate their methods experimentally. The experiment involves the tasks of grasping a hand-driller and drilling a red hole on a wooden block with the grasped drill. The authors report an average of 9.3mm drilling deviation when the sensor measurements are incorporated into the tasks and an average of 47.5mm drilling deviation without the aid of any sensor.

Hu and Xiong (2017) use a new approach, a disturbance Kalman filter, to estimate the force acting on the end effector for a compliant human-robot motion. In their approach, Hu and Xiong (2017) modified the Kalman filtering technique by using rigid body dynamic model and its disturbances as the update stage. The authors use Inverse Dynamic Model (IDM) as a prior to the force. The approach also models the sensor using rigid body dynamic. The only feedback data used as update state of their novel disturbance Kalman Filter are the joint positions and the torque measurement from the sensor. The Disturbance Kalman filter takes into account the uncertainty from the disturbance dynamics. The authors successfully implement the disturbance Kalman filter for the force estimation on a six degree of freedom Kinova Jaco2 arm robot.

Sawada et al. (2012) present a technique of collision avoidance using unscented Kalman filter for a two-link flexible manipulators. The researchers use sliding mode controller to control the motion of the manipulator. They also investigate the use of EKF to manage the uncertainty introduced by the flexible beams of the manipulator which affects the motion trajectory of the manipulator's end effector. They introduce a collision input into the observation model of the UKF. The collision is detected by a piezoelectric sensor attach at the base of the links. An abnormal reading from the piezoelectric sensors would trigger the collision input and change the parameters of the UKF to suspend control of the manipulator. This approach, regrettably, expects

collision rather than avoiding it. However, this is inevitable considering piezoelectric sensors require contact to detect changes. They confirmed that, through two numerical simulation, the algorithm for their collision detection via UKF successfully increase efficiency.

2.3 MAP-BUILDING APPROACH TO UNCERTAINTY MANAGEMENT

A robot in an industrial environment is isolated. The robot is barricaded because of its massive built which impose danger to human operator. But it is argued that with flexible workcell where a robot where a robot manipulator in the industry can manage the uncertainty of its environment through map-learning or map-building, this robot system can potentially reduce cost of operation.

I discover that there are two methods to mapping an unknown environment of a manipulator. The first method is by using point clouds Um and Ryu (2013), Wang et al. (2017). and the other by using occupancy grid map together with its variants such as octomap. I must mention that most of the researches on manipulator's mapping are not in immediate industrial setting. Yet, I am confident that these solutions can be used in industrial robot.

2.3.1 Point Clouds Mapping

Mapping with point clouds use direct data from sensor to model the environment. Often these metrical data are translated and combined from multiple scans by using iterative closest point to construct the global representation of the environment.

Um and Ryu (2013) uses point clouds to map the environment. The paper presents a simultaneous planning and mapping for a three degree of freedom manipulator in an unknown environment. The paper argues that their method handles the uncertainty of the unknown environment by using virtual skin concept. The manipulator explores the environment using a mono-vision Infrared Proximity Array (IPA) sensor which replicates the characteristic of a tactile-sensitive skin. From this explo-

ration, the robot would model its configuration space and use it as the input for their best-next-move algorithm to optimize path planning. The IPA sensors are attached on each links of the manipulator. Readers should note that an IPA sensor has limited range and noise bound output which add more uncertainty to the estimation of the manipulators configuration space. The researchers address the noise in the sensor using Group Average Feature (GAF). The pointclouds generate from the IPA go through GAF algorithm to produce a more confident map of the configuration space. The generated map of the configuration space is used to provide motion planning for the robot manipulator. The RRT path planning algorithm use the map as an input to plan the motion of the manipulator. The authors optimized the RRT algorithm with their Best Next Move (BNM) algorithm. BNM is reactive and does not use any inverse kinematic solution. They validate the method using and compare the RRT-BNM method with RRT method by simulation and showed that the RRT-BNM method has higher mapping efficiency. They defined mapping efficiency as percentage of map built compared to the actual map, over the number of point clouds in the c-space. Due to the GAF decompression method, the RTT-BNM produce map with less point clouds. The author, however, did not repeat the RRT experimentally. Despite the higher efficiency of RRT-BNM path planning, the map produce does not represent an accurate geometric information of the unknown environment explored by the manipulator. I believe this is because there is a lack of pose estimation of the manipulators link since the research disregard the use of inverse kinematics, bayesian or non-bayesian filtering techniques to reconcile with the path planning.

Wang et al. (2017) present a method using point clouds to perform self-identification using k-means clustering method and obstacle detection and obstacle avoidance. The point cloud is used to generate the skeleton and the sphere along the skeleton. The point clouds are sparse and noisy which introduce uncertainty to the self-identification. However, the uncertainty is eliminated using a filter based on the density of the point cloud. The detection of the obstacles is based on the negation of point clouds that are not a part of the skeleton and the sphere that shape the manipulator. This segmentation is the base

of their approach to obstacle avoidance. They tested their algorithm by performing three experiments sequentially using a Baxter manipulator robot with eye-in-hand configuration. These experiments involve the validation of their self-identification algorithm, obstacle detection algorithm, and obstacle avoidance algorithm.

2.3.2 Occupancy Grid Map: The Map of Uncertain Environment

Direct use of point clouds to map an environment is inefficient and requires a considerable amount of computational expense. This is not the case of an occupancy grid cell where the point clouds are used to statistically parameterized the grid cell of the map. A grid cell or a voxel is a cubic primitive that represent the environment's geometrical characteristic. Occupancy grid map consider the uncertainty of mapping sensor such as a sonar, a camera, or a laser range finder to represent the environment probabilistically (Moravec, 1989). An example of the use of occupancy grid map is by Matuszek et al. (2011) that uses occupancy grid map in presenting an environment of a chess playing robot manipulator. The construction of occupancy grid map often requires data fusion or sensor fusion where two or more sensors are fused together using Bayesian rules or any filtering techniques. Rybski et al. (2012) use a variant occupancy grid map and handle uncertainty of the environment by fusing measurements from a Swissranger SR4000's ranger and two Tyzx G3 EVS stereo cameras to estimate the occupancy of a grid cell.

I, accordingly, define this problem under the uncertainty managing the unpredictable environment unlike Janabi-Sharifi and Marey (2010) which uses the state of the robot's joint position as the visual feedback of their visual servoing). Kalman filtering technique is used to help track an object on a flat surface. The use of extended Kalman filter aside from the trajectory control of the end point of the robot increase certainty of the pose estimation and the accuracy of the tracking. A camera is mounted on the end point of CRS Plus SRS-M1A robot system. The robot is presented with a task to follow the object using the eye-in-hand camera. The visual feedback from the camera is used with their control strategy. The extended Kalman filter is used to predict the relative position of the object while it is moving. Since the researcher maintain a constant velocity

to the object, the tracking can use a Kalman filtering technique without linearization. However, since the tracking of the object requires the movement of the camera, given the nonlinearity of an inverse kinematic solution of their robot system, the use of extended Kalman filter is justified. The filter maintains the estimation of the end effector's pose. Although the robot manages to follow the moving object in their experimentation, the authors did not compare and reiterate their experimentation with other estimation method.

Burns and Brock (2007) present a method of sample-based motion planning on top of an occupancy grid map. The sampling-based planner take regard the uncertainty of sensor measurements and incorporate the uncertainty into the occupancy grid map. Their method is iterative; the planner considers and incomplete information about the world and constantly update changes in their map model and configuration space. The authors use the concept of utility and cost to weight the probability of an edge. A* algorithm is used to search the most optimal path from randomly sampled node in the configuration space. During edge validation of the roadmap is withheld before the A* algorithm complete its query. They validate their planner by simulating three separate experiments, two of which, involve manipulators with ten degrees of freedom and fourteen degrees of freedom respectively. The planner is specified to solve 50 planning queries given a start and a goal configurations. They report that building a planner on top of a occupancy grid map increase the accuracy of completing the path from the start configuration to the goal configuration. They conclude that, compared to PRM, their method able to eliminate the possibility of invalid edges in their roadmap because they introduce error model of the sensor at hand.

Kruse et al. (1996) present a method for exploring an unknown environment using either a mobile robot or a manipulator robot. Their method uses a planning-sensing-updating cycle. This research uses an occupancy grid map to represent the unknown environment. Their rating approach is similar to Burns and Brock (2007) utility and cost concept. However, they relate their rating function with the manipulator's configuration space by introducing constraint for fast exploration.

Paul et al. (2011) present a path planning strategy to perform autonomous grit-blasting as a part of bridge maintenance process. They develop a framework called Autonomous eXploration to Build A Map (AXBAM). AXBAM define the uncertainty in an unknown environment as the measurement of information that has not been discovered in the environment by using Shanon's entropy definition of information theory. This is an appropriate concept to use in modeling an unknown environment since the theory help in optimizing the exploration and path planning of a manipulator. The authors use Occupancy Grid map concept to handle uncertainty for path planning in the unknown environment based on the entropy definition. The authors implement ellipsoid force-field planner to plan collision free roadmap. Another problem addressed by the authors states that there are possibilities for multiple goal configuration and use their entropy model to arrive to a single goal configuration with the highest information gain prediction. Despite the use of occupancy map in managing the control of their manipulator under uncertain environment, the mapping model in this paper uses direct pointclouds for detailed geometric representation of the unknown environment. However, the author fails to show any optimization technique when registering local scans into a global scan. We believe without this optimization, if the pose of the end effector is uncertain, their map may have diverging misalignment. The authors use Hokuyo Laser Range Finder (Hokuyo URG-04LX) as a scanning sensor attached on a six degrees of freedom Denso VM-6093 manipulator arm with eye-in-hand configuration. The 5th joint rotates to facilitate the initial scanning of the environment before exploration commence. Based on this initial scan, the robot will start exploring. By using AXBAM, the authors claim to reduce computation in decision making during exploration.

2.3.3 Octomap: The definitive Uncertainty Management Dynamic Environment

A number of researchers represent the environment using an Octree model to facilitate path planning strategy (Faverjon, 1984; Hamada & Hori, 1996). Similar to Occupancy grid map, octree represent the environment with discrete cell that divides into smaller details. Payeur et al. (1997) use octree based representation of occupancy grid map to

reduce computational load so that the grid map can be used efficiently to represent 3D scenes.

Hornung et al. (2013) improves the use of octree probabilistically by introducing relaxed logit function for uncertainty management in mobile robots. Octomap represents the uncertainty in the environment and the sensor that maps the environment. Although Octomap has been exhaustively used in mobile platform, researchers have introduced its use in robot manipulators. By using Statistical Outlier Removal algorithm, Miseikis et al. (2017) use Point clouds from cameras to construct an environment based on octomap. They proceed by merging point cloud data from two 3D cameras (Kinect Sensor) using iterative closest point. Iterative Closest Point (ICP) is often used in mobile robot to combine two scans collected at different position together. They use forward kinematic to assist in replacing point clouds corresponding to the robot's chassis with cylindrical shapes. The map of the environment is modeled using octomap and embedded with a decaying occupancy value. The decaying cost values are used to represent danger zone, a intermediary zone, and a non-danger zone in the map for a reflexive and predictive behaviors. Building their path planning on top of these map provides a responsive motion even if the robot workspace is populated by moving objects. The authors use six degree-of-freedom (DOF) UR5 robot with eye-to-hand configuration. They validate their method by simulating a predefined back and forth motion between a start configuration and a goal configuration. The first simulation act as the baseline or the benchmark of their path planning approach using only RRT. The second and the third experiments introduce a moving object into the robot's workspace. Experiment 2 and experiment 3 use the reactive path planning and the reflexive-predictive path planning approaches respectively. They conclude that although experiment 3 performs at the shortest time, the result was not significant.

2.3.4 Miscellaneous Mapping Techniques

In this section, I present papers that have unconventional way to manage the uncertainty of their environment. Cohen et al. (2010) and Meeussen et al. (2007) use graph theory

to their mapping technique, while Petrovskaya et al. (2006) and Koval et al. (2013) interact directly with the environment using force sensor to localize object in the robot’s configuration space. Corrales et al. (2008) track a human operator in its environment. Ruhr et al. (2012) introduce the element of learning to help manipulate dynamic object in the environment. Cohen et al. (2010) present a search-based planning as an oppose to sampling-based motion planning. They use examples of motion planning in lower dimensionality problem or problems in low-dimensional manifold as a heuristic for motion planning in higher dimensional manifold. From this heuristic, they define motion primitives, a predefined motion of a single joint, and used them to minimize the cost function so that the most optimal path can be realized. This method eliminated the multiple solution to start-to-goal configuration by selecting the most feasible path that avoids collision which may not have the shortest path. Their searching-based planner follows Anytime Repairing A* (ARA*) search algorithm. ARA* is different from A* algorithm because A* always aims at getting into a goal configuration at the shortest traversal. ARA* consider a factor of the most optimal path initially which can be rectified at further path sampling. They use occupancy grid map to decrease the intractability of their algorithm so that the ARA* motion planner produces the most optimum solution in path planning. They manage the uncertainty of multiple path solution by eliminating it using cost function as constraint to their path planning algorithm. They validate their approach by simulating manipulation in a cluttered tabletop and conclude that their approach are only optimal for three degree of freedom pose (translational) rather than a full six degree of freedom pose (translational in x,y, and z directions together with orientation about the x, y and z lines) They also perform the same experimentation on a PR2 robot with seven degree of freedom manipulator.

Meeussen et al. (2007) present an approach to generate a path planning by human demonstration for a sensor based manipulator. A tool containing optical markers are tracked during demonstration phase using a 3D vision sensor (a Krypton 6D Optical System). The tool is attached to a geometrically uncertain object in a controlled environment. The use of 3D vision sensor, and the estimation of the state of the tool and

indirect estimation of force asserted by the tool based on the state of the tool introduce uncertainty. To compromise the uncertainty, the authors use particle filtering technique to estimate the pose and twist of the manipulated object via tracking of the manipulating tool the force between the contacting object. The researcher addresses their previous work on the same problem and optimize the particle filter with topological graph called contact-state graph to predict the next best configuration of the object being moved or contact formation. They observed that the contact-state graph reduces the number of particles used during sampling which decrease computational load and increase accuracy in their estimation. The paper however did not replicate their simulation in an experimentation with an actual manipulator

Petrovskaya et al. (2006) use particle filter to estimate the position of their end effector and at the same time ascertain the position of an object by using tactile sensor. The sensing by touching follows a heuristic where a person increases its confident of the shape and the position of an object by feeling it with his or her hand at different location. From this heuristic, the authors express their algorithm with the help of particle filtering technique. Particle filter depends on sampling approach where the prior or a belief is increased by each successive sampling. They modified the particle filtering technique to assist a highly sparse measurement from tactile sensor and called their technique as Scaling Series Particle Filter (SSPF). In SSPF, the initial filtering steps has the lowest resolution of accuracy and the highest uncertainty. Based on the initial sampling, the following sampling increase increase the certainty of the position of an object by sampling at a different position. The authors tested their algorithm by simulation and experimentation. In the experimentation, SSPF manage to help a manipulator localize and grasp a box at 70% success rate, and identify and handle a door knob at 98% success rate.

Koval et al. (2013) present a situation where constant contact and manipulation of an object, by pushing the object, can be used to estimate the state of the object; i. e. the position and the orientation of the object. They term the process as contact manipulation. They argue that with the absence of more conventional observation from sensors

such as a laser range finder and a vision camera, tactile based sensor can perform state estimation using appropriate filtering technique. Since tactile sensor has non-gaussian characteristic and highly nonlinear, the authors used particle filter as their estimator. The use of tactile sensor, however, introduce uncertainty because the sensor has low spatial resolution or low manifold. This is regarded as a low-dimensional manifold problem where the state estimation has higher dimensionality (two for position on a plane and one for orientation). Although intuitively increasing the resolution of the tactile sensor may decrease the spatial uncertainty, it is not the case for tactile sensing with particle filtering technique. Thus, the authors introduce Manifold Particle Filter (MPF). An MPF reduce the dimensionality of the state estimation by marginalizing the probability distribution of the state of the object based on the observation from the tactile sensing and use it as a prior estimate. To implement the MPF, the state estimation follows three steps:

- (a) assumption of the state of the object by evenly weighting particles
- (b) action which involve pushing the object
- (c) observation where the MPF use the pressure profile during pushing to estimate the state of the object.

They implement the algorithm using OpenRAVE simulation environment and evaluate it with a simulated BarrettHand. They then run an experiment using Andy, a robot module developed for Darpa ARM-S competition. They compared their result with a Conventional Particle Filter (CPF) and improved estimation of the object state.

Corrales et al. (2008) uses kalman filter fusion algorithm to fuse Ultra-wideband (UWB) technology and inertial motion capture system to estimate the motion of human operator in industrial environment. The algorithm improves the interaction between a robot and a human operator/user by localizing a human in a manipulator's workspace so that a cooperative interaction can be made. Since the authors use UWB sensor to estimate the location of the human, they use Kalman filtering technique to fuse the

information coming from the UWB and the inertial sensors. The filtering technique compensate the low data rate of the UWB and the high error from inertial sensor. Their Kalman filter uses the global position of the UWB as the correction step and the inertial sensor as the prediction step. Their result shows that by fusing two measurements using kalman filter algorithm, the state estimation of the person's location is increased in accuracy.

Ruhr et al. (2012) This paper present solution to manipulation task which involve opening and closing doors and drawers in any kitchen environment. Their approach involves the management of uncertainty of door handling through learning. Within their learning model framework, the authors use 3D point clouds directly to identify door or cabinet handles. They use Random Consensus (RANSAC) with the point clouds to segments the point clouds to help detect the handles based on identification of planes that parameterized a wall, ceiling, floor, and cabinets. They also implement real-time impedance control and kinematic model learning to estimate the kinematics of dynamic objects in the kitchen. They use a higher level of abstraction in representing the environment via semantic maps. They evaluated their approach using PR2 mobile articulated robot which has seven DOF manipulator. They report out of 104 trials of opening and closing cabinets and doors, the rate of success is 51.9%.

2.4 SIMULTANEOUS LOCALIZATION AND MAP-BUILDING AS A TOTAL SOLUTION TO UNCERTAINTY IN A MANIPULATOR'S STATE AND ENVIRONMENT

I have discussed the uncertainty of a manipulators can be represented and manage using filters. I have also discerned the use of occupancy grid map, point clouds, and octomap to model the environment of a manipulator. Despite rich solution option to uncertainty of a robot state and its environment, the solutions are disjoint and performed separately.

The closest solution for SLAM problem in a manipulator robot is reported by Klingensmith et al. (2016). The authors argue that encoders on each joints of a manip-

ulator is not enough to estimate the endpoint of a manipulator due to gear trash, cable stretch non-rigid deformation and others. Their Articulated Robot Motion for Simultaneous Localization and Mapping (ARM-SLAM) uses Truncated Signed Distance Field (TSDF) as part of the scene reconstruction to help estimate the pose of the end effector. TSDF is a variant of Dense Fusion that performs 3D scene reconstruction using multiple depth images and camera poses. With TSDF, each voxel is encoded with a distance to the nearest surface. The voxels are weighted where positive weight means the voxel is outside the surface, negative weight means the voxel is inside the surface and zero distance is when the voxel is on the surface. Their ARM-SLAM adopt eye-in-hand configuration. They conducted three experiments to validate their SLAM solution. In the two-dimensional simulation, they compare pose error between forward kinematics, Dense Fusion algorithm and ARM-SLAM algorithm. The result shows significant error reduction for both of the two algorithms compared to the forward kinematics calculation. ARM-SLAM, however, has the lowest errors. In their 3D simulation experiment, the authors compare results for forward kinematics calculation with Kinect fusion and ARM-SLAM. They observed that ARM-SLAM is more robust when loss of data occurs. They also conducted a real shelf scanning. In this experiment, the authors could not conclusively see better pose estimation compared to the forward kinematics calculation. However, they restate that during loss of data, the ARM-SLAM solution produces robust estimation.

Based on Klingensmith et al. (2016), I conclude a SLAM approach represent the total uncertainty of a manipulator in an environment. I characterize the SLAM problem as the methodology that involve the consideration of robot's state uncertainty, its environment uncertainty, and a tractable solution that uses both the former and the latter. In a manipulator, a SLAM problem can be considered as a model that has tractable solution given the uncertainty of its end effector's pose and the uncertainty of its configuration space. We define that the SLAM solution to a manipulator should have an element of map-learning where statistical and probabilistic approach is considered when estimating the pose of an end effector.

Sun et al. (2016) use Large-scale Direct mono-SLAM (LSD-SLAM) technique to replicate the motion of a human operator. The technique resolves the noise from their measurements using dbscan, a density-based spatial clustering algorithm, to eliminate outlier. LSD-SLAM uses a visual camera to produce point clouds with the help of db-scan algorithm. The scenes generated from the camera are three dimensional. However, the movement of the manipulator is planar. The LSD-SLAM is used to replicate the movement of a human arm. The movement are learned and modeled using Gaussian Mixture Model (GMM) and the parameters of the model are estimated using Gaussian Mixture Regression (GMR). GMM is a probabilistic model that assume all the data points are generated from a mixture of finite number of Gaussian distribution with unknown parameters. GMR uses Expectation maximization iterative learning algorithm to help replicate an output data based on an input. I observe that, by using GMM and GMR, the authors manage to handle the movement uncertainty of the human operator and map the movement into a planar actuation of the manipulator. The authors perform an experiment by recording the movement of the human operator. The LSD-SLAM manage to generate a smooth path for the end effector of the manipulator as prescribed by the GMM and GMR based on the demonstration of the operator.

Nissler et al. (2016) solution to AprilTag tracking is similar to SLAM solution method where AprilTag is used as a feature observed by the measurement to optimize the pose estimation of the end effector and at the same time using the pose estimation to track the AprilTag on a work space. Nissler et al. (2016) demonstrate the use of fully autonomous handling of CFRP material. The authors use RANSAC solution to fuse multiple scene from sensors, an eye-in-hand camera, that may contain outliers. The authors use KUKA KR 210 with a AVT GigE camera attached to the end point. The robot follows three different motion; a horizontal arc, a vertical arc, and an approach motion. A laser tracker is used to calibrate the position of the robot and the markers. A single AprilTag marker was tracked during the motion. The motions were repeated while tracking multiple AprilTag markers. RANSAC method was compared with a least square method. They observed that RANSAC method that they developed for the

AprilTag tracking improves the estimation of the end effectors pose.

I also noted that SLAM solution is used for mobile manipulators. Song et al. (2013). This paper present two problems and two solutions. One is on self-localization, a SLAM problem which they solved using EKF-SLAM and the other is on grasping problem. I will only regard the grasping problem since it directly involves manipulation under uncertainty. They use Speed Up Robust Feature (SURF) for object identification and visual servoing using Position-based Visual Servoing (PBVS) to grasp an object and return the object to a person. During object identification, the authors raised the concern of matching error during identification which may result in misleading object recognition. They mitigate the problem using random sample RANSAC to reject outliers and to find the homography matrix. Homography matrix transforms points in one image to the corresponding points in another image when a camera changes in position. The result of their investigation suggests a successful object retrieval to a person. They did not, however, present any comparison of their result with other approach for the grasping problem.

From these papers, I observe that the having to mount a manipulator to a mobile platform introduce localization uncertainty where there is uncertainty on the base location of the manipulator arm, hence propagating the uncertainty to the path planning of the manipulator.

Pilania and Gupta (2015) propose a method of base uncertainty by using their novel Hierarchical and Adaptive Mobile Manipulator Planner Uncertainty (HAMP-U) to account for the uncertainty of the base of a mobile robot

I also observe manipulator attached to a mobile platform often regard their grasping solution using two different approach; i.e. a probabilistic approach for the mobile platform localization and a deterministic approach at the manipulation stage of the robot movement such as grasping (Gasparri et al., 2006; Venator et al., 2013). I also note that the total uncertainty solution presented by SLAM does not extend to path planning approach.

2.5 THE PLANNING FOR 6R ROBOTS IN DYNAMIC ENVIRONMENT

This section delves into a small number of research papers into motion planning in a dynamic environment for robot manipulators in three dimensional space, \mathbb{R}^3 . We pay close attention to algorithms derived from sampling-based planner which use stochastic approach to query the configuration space. The planning algorithms for robot motion in a dynamics environment dated back to 1985, albeit mainly applied on mobile robot (Mohanan & Salgoankar, 2018).

Some planning algorithms that is non-probabilistic for robotic arm are also observed which are based on representation space (Liu et al., 2016; Su & Xie, 2011), sequential expanded Langrangian homotopy (Dharmawan et al., 2018), and Real-time Adaptive Motion Planning (RAMP) (Vannoy & Xiao, 2008). These works provide a new method of planning and tackle the problem at the local planning and at lower level of control to solve problems with the aid of a simulated environment. Their simulated environment, or a map model of the environment, includes dynamic objects that are then filtered and disregarded in the map registration pipeline. Their experimentation approach will be adopted in this research where, the map is informed with the presence of obstacles that are moving in the workspace without having another sub-module of the system tracking the object via motion tracking.

2.5.1 The Probabilistic Motion Planning

Kavraki et al. (1996) is the first group of researchers that used probability model for sampling the configuration space for holonomic robot motion such as a manipulator robot. The planner are called The Probabilistic Roadmap (PRM) motion planning. The algorithm construct a graph structure to find path between an initial pose to a goal pose in two-dimensional configuration space, $n = 2$. Kavraki et al. (1996) also proof a more general solution for higher dimensional configuration space, $n > 2$. With graph structure, more than one path connect the initial pose to the goal pose. Therefore, PRM is a multi-query type planner.

Kunz et al. (2010) improve PRM by redefining the distance metric of a robot manipulator so that the robot can move around a moving obstacle in real-time. Their approach performs well in an uncluttered environment. Kunz et al. (2010) also redefined the distance function of the PRM to address dynamic objects, such as a walking person, into a two-dimensional map. Although the configuration space of the manipulator is in \mathbb{R}^3 , the map , constructed from a two-dimensional Laser Imaging, Detection and Ranging (LiDAR) scan, is in \mathbb{R}^2 .

In retrospect, the RRT was formulated for non-holonomic motion (LaValle, 1998) targeting problems addressed in differential-constrained motion such as a car on a plane. However, given the model of its metric space and consequently the configuration space, RRT are tractable for higher dimensional problem such as manipulator motion in 3D space (Wei & Ren, 2018). RRT assume as static environment but Wei and Ren (2018) successfully change the way RRT samples a robot metric space so that it is fast enough to react with a changing environment. Also, unlike PRM, RRT works well in a cluttered environment because of the randomized sampling on the robot configuration space in the metric space.

Researchers have been modifying the PRM (Jailllet & Siméon, 2004; Klasing et al., 2007; Likhachev et al., 2005; Pomarlan & Sucan, 2013) and the RRT (Bekris & Kavraki, 2007; Ferguson et al., 2006; Ferguson & Stentz, 2007; Otte & Frazzoli, 2015) to facilitate better performance. Unlike Kunz et al. (2010) and Wei and Ren (2018), so few have applied their planning algorithms on a robot manipulator despite both algorithms provides mathematical framework for .

We will use the method demonstrated by Kunz et al. (2010) and Wei and Ren (2018) to design our experiment of a moving obstacle collision avoidance with the implementation of the vanilla RRT to solve motion for robot manipulator in three-dimensional space, \mathbb{R}^3 . Different from the implementation by Wei and Ren (2018) our method implement the vanilla RRT where we do not represent the obstacle configuration space.

2.6 SUMMARY

It is concluded that uncertainty management of a manipulator is imperative for a flexible system in an industrial environment. The uncertainty of the manipulator is best approached by considering the uncertainty motion of the robot and the uncertainty of its environment. This thesis has reviewed research and works that manage the uncertainty of both categories separately. A number of researches that incorporate both uncertainties into one framework by using simultaneous localization and mapping applied on robot manipulator is small in number. All of these uncertainties management involve the use of statistical and probabilistic approach. However, no definitive Planner-SLAM solution has been discovered thus far in this review. Referring back to the problem statement of this thesis, a tractable solution that incorporate the uncertainty of the robot's state, and the dynamic environment via a Planner-SLAM solution remains unreported. Accordingly, this thesis will attempt to bridge the gap for a reconciled planner-SLAM solution into a pipeline in the following chapters.

CHAPTER 3

MATHEMATICAL BACKGROUNDS

The role of this chapter is to provide the primer for the mathematical foundation of this thesis.

A six-axis robot with six revolute joints has been developed as the hardware platform to validate the Planner-SLAM solution proposed in this thesis. The robot has been named Richard Mini (code as *r_mini*). This chapter presents the features and mathematical model of this robot.

The chapter revolves around the geometric representation of the kinematic model of *r_mini* and the mathematical framework for the planner used in this research and the components of the SLAM solution.

3.1 THE KINEMATICS OF RICHARD MINI

3.1.1 The Mechanical Descriptions

Richard Mini (*r_mini*) has one fixed link, six moving links, six revolute joints, and six frames attached to the links. The links are named numerical: link1, link2, ..., link6 shown in figure 3.3. The same is true for the joints: joint1, joint2, ..., joint6. All joints are described by the frames attached to the links. In figure 3.2, the frames' orientations are illustrated. These joints represent the state of the manipulator which parameterized the control space $\{c^n \in C^n\}$. C^n for *r_mini* is a 6-dimensional control space.

This thesis will use the superscript notation to refer the control space set and its elements and the subscript as the equivalent notation in the configuration space. For example, C^{ee} , refers to the control space at the end-effector where the controlling pipelines for *r_mini* would take in $c^{ee} = (\theta_1, \dots, \theta_6) \in C^6$, and the equivalent pose is in the con-

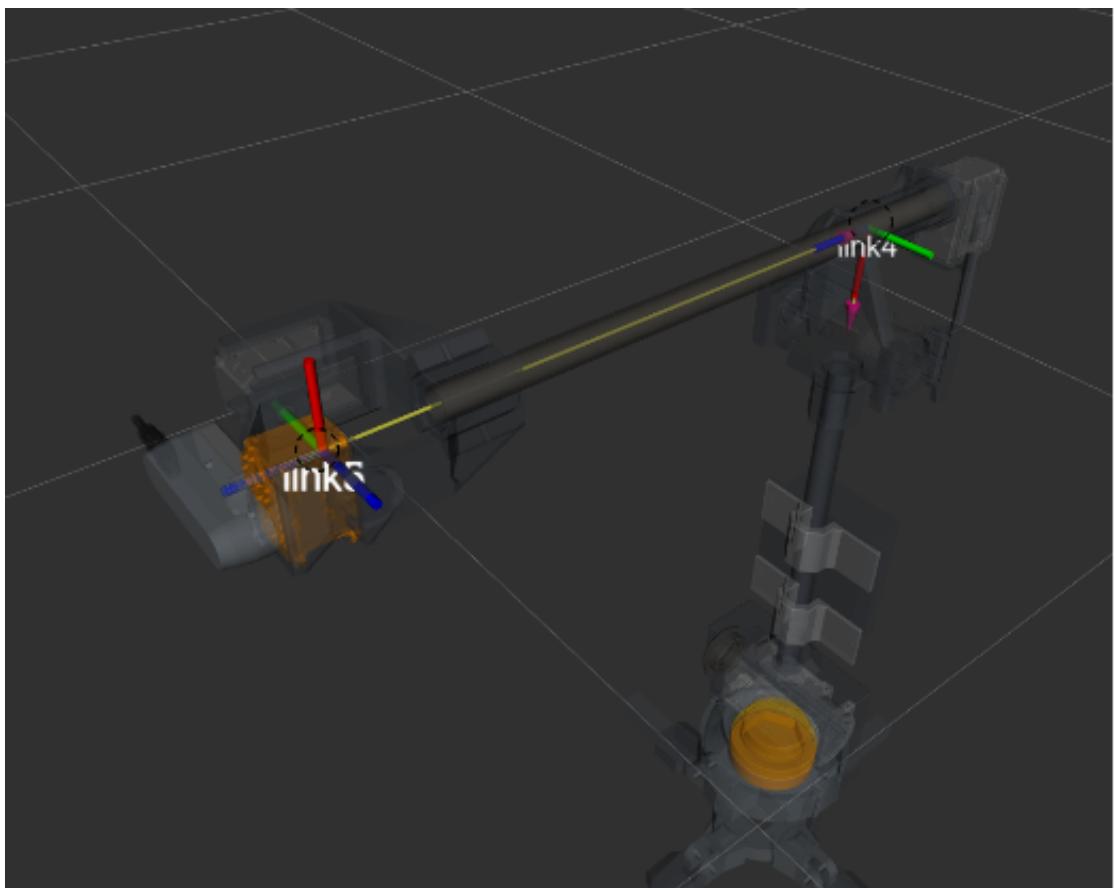


Figure 3.1: *r_mini* wrist conforms to Pieper condition where axis of rotation for joint4, joint5, and joint6 share points of intercept. The dash circle in the diagram a possible point of intercepts. Both point are valid for a Pieper condition

figuration space, C_{ee} . r_mini follows the condition advised by Pieper (1968) :three last joints are collated and have a shared rotation axis point as prescribed in figure 3.1.

The figure 3.2 shows the $z - \text{axis}$ of the frames aligns with the rotation of the actuators on the robot.

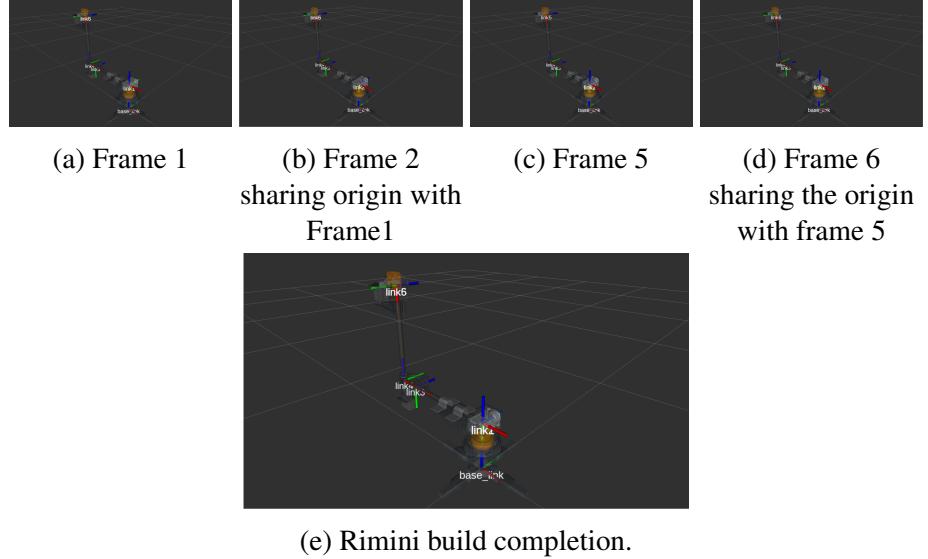


Figure 3.2: The location and orientation of r_mini . The choice of the orientation for each frames are based on Denavit-Hartenberg. The joints are values represented by the angle between two $x - \text{axes}$ around the $z - \text{axis}$ or rotation axis of each actuators

These three joints represent the wrist. r_mini also uses $Z - Y - Z$ Euler convention to present rotation state of the end effector. This convention conforms to the Wigner-D matrix parameterization where this research will use to estimate the orientation of the end-effector. However, the forward and inverse kinematic computation uses the quaternion.

3.1.2 Forward Kinematics of r_mini

The kinematics of r_mini follows the Denavit-Hartenberg (DH) convention. In this convention, the homogenous transformation of one frame to the next frame in the kinematic chain of the robot does not entails any rotation about the $y - \text{axis}$. In so doing, the homogenous transformation between two frames are simplified. By disregarding the rotation about the $y - \text{axis}$, matrix-to-matrix multiplication between two frames required for homogenous transformation is computationally relax. To help satisfy this constraint

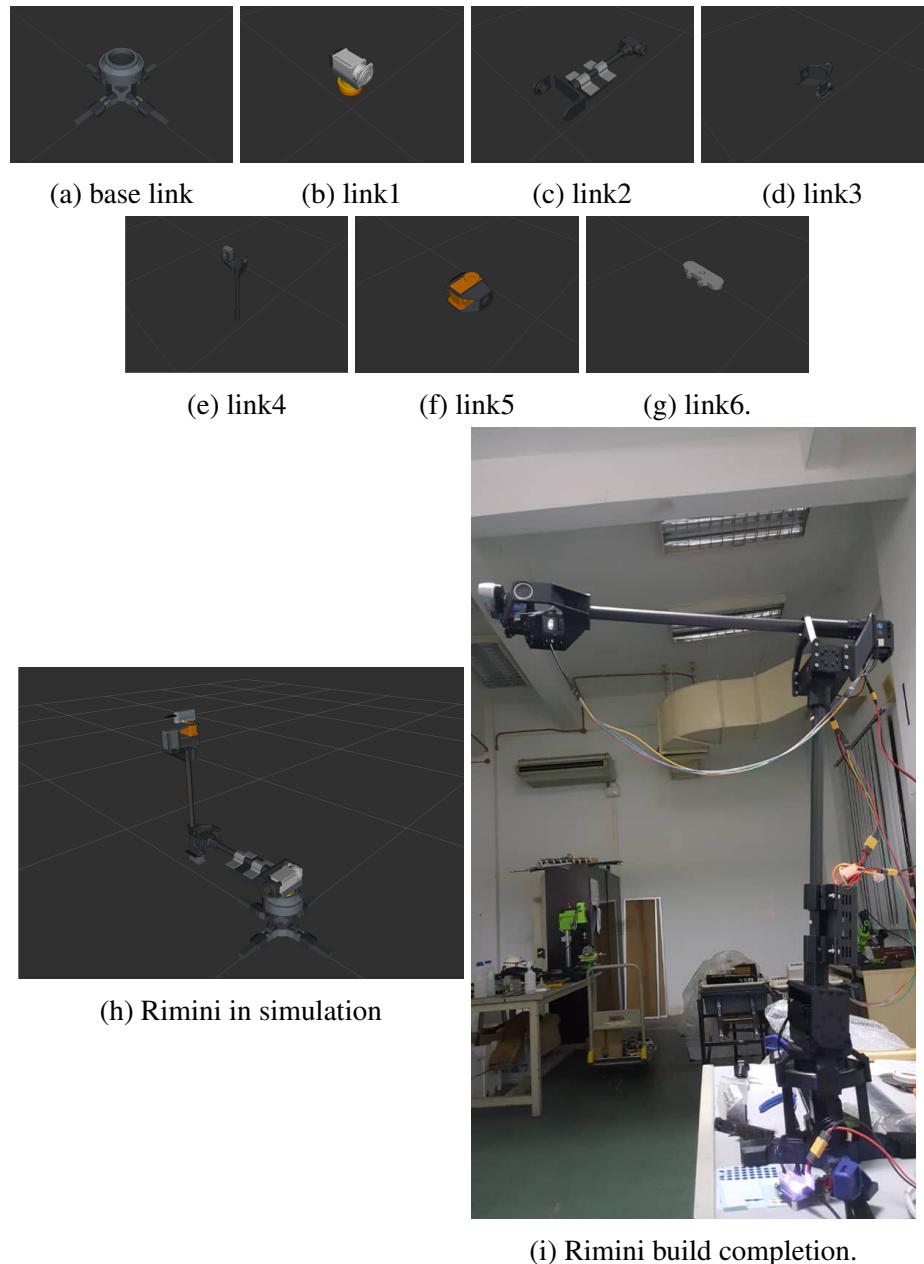


Figure 3.3: *r_mini* and its links

Table 3.1: DH-parameter table

Link (i)	a_i	α_i	d_i	θ_i
1	0	0	0.196	θ_1
2	0	-90°	0	θ_2
3	-0.373	0	0	θ_3
4	-0.08	-90°	0	θ_4
5	0	-90°	0.391	θ_5
6	0	-90°	0	θ_6

about the $y - \text{axis}$, two condition should be followed:

- The $x - \text{axis}$ belonging to $frame_i$ should be perpendicular to the $z_{i-1} - \text{axis}$ belonging to the $frame_{i-1}$.
- The $x_i - \text{axis}$ intersects the $x_{i-1} - \text{axis}$

Here, the subscript i represent the frame number which extend to the link and joint number. Based on these constraints, the DH parameters are summarized in table 3.1, where a_i is the offset of the $frame_i$'s $z - \text{axis}$ along the $x - \text{axis}$, α_i is the rotation of $frame_i$ about the $x_i - \text{axis}$, d_i is the offset of the $frame_i$'s $x - \text{axis}$ along the $z_i - \text{axis}$, and θ_i is the $joint_i$ angle of rotation. DH-convention states that each joint values are defined by the angle between two $x - \text{axes}$. As an example joint1 is represented by the angle between frames attached to link1 and the base frame. Similarly joint2 is represented by the angle between x_1 and x_2 axes.

Each row in table 3.1 describes the homogenous transformation, A_i from the $frame_{i-1}$ to $frame_i$. The transformation is repesented by four matrix operation:

$$A_i = Rot_{z,\theta_i} T_{z,d_i} T_{x,a_i} Rot_{x,\alpha_i} \quad (3.1)$$

where $Rot_{axis,angle}$, and $T_{axis,angle}$ are the representation of rotation and translation in special Euclidean group $SE(3)$. Here $A \in SE(3)$. The homogenous transformation is represented by the homogenouns transformation matrix defined as:

$$A = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \quad (3.2)$$

where $R \in SO(3)$ and $t \in \mathbb{R}^3$. $SO(3)$ is the special orthogonal group that represent orientation of the frames in the 3D space.

With each row of the DH-parameter table 3.1 represented by a homogenous transformation matrix A , r_mini configuration space is mapped by the forward kinematic function $f_{FK} : C_n \rightarrow \mathbb{R}^3$:

$$\begin{aligned} c_n &= f_{FK}(c^n) \\ &= A_1^0(\theta_1)A_2^1(\theta_2)A_3^2(\theta_3)A_4^3(\theta_4)A_5^4(\theta_5)A_6^5(\theta_6)p_0 \end{aligned} \quad (3.3)$$

3.1.3 Inverse Kinematics of r_mini

The inverse kinematics solution for r_mini follows the analytical approach derived by Pires (2007), since both the IRB1600 and r_mini has the Pieper condition addressed in section 3.1.1. However, given that the analytical approach introduce multiple solution, accordingly, subjected the motion planning to singularities, this report adopts a numerical method approach which is used during the simulation and experimentation.

Newton-Raphson method to find the inverse kinematic solution of r_mini , \hat{c}^{ee} which is the estimate of c^{ee} . The generalization of the method uses the current value of the robot's encoder, $c^{current}$, and the termination value, $\epsilon = 0.005$, to end the iteration. Algorithm 1 delineate the the method.

3.2 R_MINI 'S PATH PLANNER

This research uses RRT implementation provided by OMPL library packaged in the MoveIt software. The algorithm for the purpose of this research is shown in *algorithm 2* where k represent the, number of node in the tree generated by the RRT, \mathcal{M} , represent the collision map which is part of the planning scene where all RRT sampling takes

Algorithm 1: getInverseKinematics

Input: $c_{goal}, c^{current}, \varepsilon$
Output: \hat{c}_{ee}

```
1  $e \leftarrow \text{getForwardKinematic}(c^{current})$ 
2 while  $\|e\| \neq \varepsilon$  do
3    $\hat{c}^{estimate} \leftarrow c^{current} + \text{getInverseJacobian}(c^{current})e$ 
4    $e \leftarrow C_{goal} - \text{getForwardKinematic}(c^{estimate})$ 
5    $c^{current} \leftarrow c^{estimate}$ 
6  $\hat{c}^{ee} = c^{current}$ 
7 return  $\hat{c}$ 
```

Algorithm 2: generateRRT

Input: $C_{initial}, C_{goal}, \Delta t, k, \mathcal{M}$
Output: \mathcal{T}

```
1  $\mathcal{T}.\text{initialize}(C_{initial}, C_{goal}, \mathcal{M})$ 
2 while  $c_{new} \neq C_{goal}$  do
3    $c_{random} \leftarrow \text{randomState}()$ 
4    $c_{near} \leftarrow \text{kNearestNeighbor}(k, c_{random}, \mathcal{T})$ 
5    $u \leftarrow \text{selectInput}(c_{random}, c_{near})$ 
6    $c_{new} \leftarrow \text{newConfiguration}(c_{near}, \Delta t)$ 
7    $\mathcal{T}.\text{append}(c_{new}, c_{near}, u)$ 
8 return  $\mathcal{T}$ 
```

place and \mathcal{T} is the tree that points to a non-colliding space. In this RRT implementation, the map are loaded or queried in line 1 each time the *generateRRT()* is invoked. Line 3 generates a random state bias towards the C_{goal} . Line 4 invokes the k-nearest neighbor to find a selection of nodes that is closes to the state configuration, c_{random} . Line 5 is the important part of the sampling in the RRT where it represent the controlling input of the robot motion. Since, the robot are control in the joint-configuration space, the angular joint limit address the shape of the workspace. However, given the angular velocity, these limits are translated into the configuration space via the kinematic Jacobian which requires the information on the Δt , a tree node searching constraint. The limits implicitly ensures that the RRT, by executing line 5 within the context of the robots Jacobian, does not pass through the singularities of the robot. Hence, the configuration space of the manipulator also includes, C_{limit} , containing configuration that abides the joint-control space ranges and angular velocity limits.

The configuration space where the RRT-sampling is concerned is modified in this thesis where, the rotation representation and its sampling is in $R \in \mathbb{H}$, such that the parameterization of the Hamiltonian-space is the quaternions 4-vector, $q \in \mathbb{R}^4$. Therefore, the representation of the robot poses and also the non-colliding poses, (q, t) are explained in *equation 3.4*.

$$c_{pose} = \begin{bmatrix} q \\ \dots \\ t \end{bmatrix} \quad (3.4)$$

The RRT sampling involves query into a map, that stores collision objects. This is the planning scene, denoted as collision map, where the RRT sampling occurs. The query are called when both initial pose and a goal pose are sent to the RRT planner input. The output of the pipeline is a set of non-colliding space where from there another pipeline, transform the configuration space into a control space. The control space is defined in the following section.

3.2.1 The Cycle Space

The cycle space is the subset of the planner solution where the RRT algorithm is invoke twice. During the generation of the cycle space, the RRT output the a trajectory from the initial pose, $c_{initial}$ to the goal pose, c_{goal} , into a controlling pipeline. The trajectory are then map from the configuration space into the joint-configuration space via the Newton-Raphson inverse kinematic solver (algorithm 1). To complete the set of the cycle space consist of trajectories, τ , the entries in the initial pose and the goal pose are swapped, while invoking query into the collision map,

$$\tau = (C^{control}, t) \quad (3.5)$$

which forms a cyclical motion between the initial pose and the goal pose. Here,
algorithm 3 line 4 algorithm 3 line 7
 $C_{cycle} = \overbrace{C_{initial \rightarrow goal}}^{\text{algorithm 3 line 4}} + \overbrace{C_{goal \rightarrow initial}}^{\text{algorithm 3 line 7}}$. Algorithm 3 block explains how the C_{cycle} space is constructed.

Algorithm 3: generateCycleSpace

```

Input:  $c_{initial}, c_{goal}, \Delta t$ 
Output: success_status

1 Function generateCycleSpace( $c_{goal}, c_{initial}$ ):
2   success_status  $\leftarrow$  false
3   while within_iteration do
4      $\mathcal{T}_{initial \rightarrow goal} \leftarrow$  generateRRT( $c_{initial}, c_{goal}, \Delta t$ )
5     success_status  $\leftarrow$  moveRobot( $\mathcal{T}_{initial \rightarrow goal}, \Delta t$ )
6     if success_status = true then
7        $\mathcal{T}_{goal \rightarrow initial} \leftarrow$  generateRRT( $c_{goal}, c_{initial}, \Delta t$ )
8       success_status  $\leftarrow$  moveRobot( $\mathcal{T}_{goal \rightarrow initial}, \Delta t$ )
9   return success_status

10 Function moveRobot( $\mathcal{T}, \Delta t$ ):
11   for all index in  $\mathcal{T}.vertices$  do
12      $c^{cycle}(\text{index}) \leftarrow$  getIK( $\mathcal{T}.\text{vertex}(\text{index})$ )
13      $t \leftarrow \mathcal{T}.u.(\text{index})\Delta t$ 
14      $\mathcal{T}.\text{append}(c^{cycle}, t)$ 
15   success_status  $\leftarrow$  TrajectoryController( $\tau$ )

```

The control space are represented by the trajectory in the joint-control space

$C^{control} \subset C^{cycle}$. In *equation 3.5* the joint-configuration space are equivalent with the configuration space in *equation 3.4*. The control space is the direct controlling parameters for the movement of r_mini where it only handles the control space (or joint-state space). The sampling of the RRT to generate the tree data structure, \mathcal{T} , are done within the $\mathbf{SO}(3) \times \mathbb{R}^3$ topology. The free configuration space, or the non-colliding pose, are represented by, $C_{free} = C_{workspace}/C_{obstacle}$. According to LaValle (1998), the $C_{obstacle}$ also covers the physical constraint of the non-holonomic movement of the robot. However, in the case of an articulated robot arm in this research, the configuration limitation are the range of the joints and the angular velocity limit. Since, all of these measurements are in the n-hyperspace, to map them into the $SO(3) \times \mathbb{R}^3$, we use the kinematic Jacobian.

3.3 THE SPHERICAL HARMONICS

In this thesis, spherical harmonics are used to discern the estimate of the orientation of r_mini 's end effector via the raw data coming from the RGB-D sensor. The spherical analysis derives from the fourier transformation of non-periodic function. Here, we will represent the pointclouds coming from the RGB-D sensor as an unknown functions (Osteen et al., 2012). With spherical harmonics, the state of the end effector are computed based on two overlapping functions \tilde{f} and \tilde{g} both representing the pointclouds. These functions are projected into a unit-hypersphere $S^2 = \{x \in \mathbb{R}^3 \mid \|x\|_2 = 1\}$. where the projection of \tilde{g} and \tilde{f} based on the same parameterization in Healy et al. (2003). Convolution of the two function are represented by inner product, acting on the Hilbert space, $L^2(S^2)$:

$$\langle \tilde{g}, \tilde{f} \rangle = \int_{\omega \in S^2} \tilde{g}(\omega) \overline{\tilde{f}(\omega)} d\omega \quad (3.6)$$

where $d\omega = \sin(\theta)d\theta d\psi$ and \tilde{g} , \tilde{f} are the arbitrary integral functions on S^2 and $\overline{\tilde{f}}$ is the complex conjugate.

The spherical Fourier transform of any function $\tilde{f} \in S^2$ is defined in $\tilde{F}_m^l = \langle \tilde{f}, Y_m^l \rangle$,

where Y_m^l are the spherical harmonics of degree $l \in \mathbb{N}_0$, and of order $m \in [0, l] \in \mathbb{N}_0$. This forms the orthonormal basis over $L^2(S^2)$,

$$Y_m^l(\omega) = (-1)^m \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_m^l(\cos(\theta)) \exp(-jm\psi) \quad (3.7)$$

where P_m^l are the associated Legendre polynomials implemented by Healy et al. (2003).

The integral over the sphere are evaluated using the sampling theorem formulated by Driscoll and Healy (1994):

$$F_m^l = \frac{\sqrt{2\pi}}{2\tilde{B}} \sum_{j=0}^{2\tilde{B}-1} w_j \tilde{f}(\theta_j, \psi_k) \bar{Y}_m^l(\theta_j, \psi_k) \quad (3.8)$$

,

where \tilde{B} is the spherical bandwidth, θ and ψ are the samples with the corresponding weight w . Equations 3.8 gives us a way to transform the function \tilde{f} and \tilde{g} which belongs in \mathbb{R}^3 into the Hilbert space, i.e. the frequency domain, via this relationship:

$$\tilde{f}(\omega) = \sum_{l \geq 0} \sum_{m \leq \pm l} \tilde{F}_m^l Y_m^l(\omega) \quad (3.9)$$

The implementation of the equations 3.7 3.8 and 3.9 are coupled with the data-structures of in equations 3.17 3.19 3.20 and 3.21 to form a correlation between the two overlapping functions $\tilde{g} \in L^2(S^2)$ and $\tilde{f} \in L^2(S^2)$:

$$\tilde{C} = \sum_{l>0} \sum_{m \leq l} \sum_{m' \leq l} \tilde{F}_m^l \overline{\tilde{G}_{m'}^l D_{m'}^l} \quad (3.10)$$

$$\tilde{C}_r = \sum_{l>0} \sum_{m \leq l} \sum_{m' \leq l} \tilde{F}_{m,r}^l \overline{\tilde{G}_{m',r}^l D_{m'}^l} \quad (3.11)$$

$$\tilde{C}_g = \sum_{l>0} \sum_{m \leq l} \sum_{m' \leq l} \tilde{F}_{m,g}^l \overline{\tilde{G}_{m',g}^l D_{m'}^l} \quad (3.12)$$

$$\tilde{C}_b = \sum_{l>0} \sum_{m \leq l} \sum_{m' \leq l} \tilde{F}_{m,b}^l \overline{\tilde{G}_{m',b}^l D_{m'}^l} \quad (3.13)$$

where \tilde{C} represent the correlation for the range encoding, and \tilde{C}_r , \tilde{C}_g , and, \tilde{C}_b represent the correlation for the red, blue, green, band images. The function D , the Weigner-D matrix, helps to rotate the spherical harmonics in the Hilbert space. The function is parameterized by the Z – Y – Z Euler-angles rotation. Since, the end effector are parameterized with the same orientation, the orientation estimate of the robot's end effector are given by the argument max of the resultant correlation,

$$r(\theta_4, \theta_5, \theta_6) = \underset{r \in SO(3)}{\operatorname{argmax}} (\tilde{C} \boxplus \tilde{C}_r \boxplus \tilde{C}_g \boxplus \tilde{C}_b) \quad (3.14)$$

where \boxplus represent Gaussian data fusion technique.

3.4 THE USE OF KINEMATICS JACOBIAN AND THE SINGULARITIES OF THE MANIPULATOR TO RELAX THE COST FUNCTION

The kinematic Jacobian is defined as,

$$\dot{c}_{ee} = J^{-1} c^{ee} \quad (3.15)$$

where given the right-hand side of the equation, angular velocity in any of the joints changes abruptly when the determinant of J approaches zero. This abrupt changes is not just because of the zero-determinant of the Jacobian, J , but also a consequence of the duality in the solution explained in section 3.1.3. At a point where the determinant of the Jacobian of the fundamental kinematics equations approaches to zero, the angular velocity of any of the joint state approaches to infinity. This behavior is represented by

the mapping of the configuration space to the end effector's state.

The fundamental kinematics equation demonstrate a peculiar characteristic. The equation 3.15 explains, regardless of the infinity velocity when approaching singularity, the task space does not reflect any abrupt or catastrophic movement. This behavior when discern in context of kinematic equation 3.15, shows that the robot configuration changes abruptly when traversing into a subset of the workspace that coincide with zero-determinant Jacobian. Should a closed-form solution to the subset of the function that represent the singularity in the workspace is needed, readers can venture into invoking the fundamental theory of determinants that give constraints and conditions to satisfy non-invertible Jacobian. However, caution ensues as this efforts may not be trivial if solved analytically. However, since the sampling space for the RRT includes the kinematic constraint and also the velocity constraint of the robot arm, the solution from the RRT, and therefore, the trajectory of the robot will never cross the singularity region. This implies that, the determinant of the Jacobians, when RRT sampling considers the kinematic and velocity constraint of the robot, will never approach zero.

3.5 SLAM FOR *R_MINI*

r_mini's end effector is an RGB-D camera: the Intel Realsense 435D. The camera livefeeds point clouds, imagery, and a calibrated pointcloud imagery where each point are encoded with RGB-value; a depth image. The depth images are bound by noise. Hence, the mathematical model of the map are based on Bayesian probability model:

$$P(n|z_{1:t}) = \left[1 + \frac{1 - P(n|z_t)}{P(n|z_t)} \frac{1 - P(n|z_{1:t-1})}{P(n|z_{1:t-1})} \frac{P(n)}{1 - P(n)} \right] \quad (3.16)$$

where $P(n)$ are the prior probability distribution, $P(n|z_{1:t-1})$ are the previous estimate of the map and z_t is the measurement posterior distribution, and n , a node in a segmented space; the map model encoding of a octree datastructure. This maps impute the probability value into the segmented space, representing occupied and free

space. To relax computation, the continuous values are conditioned to encode binary values by transforming the domain of the random variables n into a logarithmic scale using logit model. The numerator in equation 3.16 are normalizing parameter which can be disregarded without having to change the mean, mode, and the median of the maps distribution (Hornung et al., 2013). Hence, together with the relax logit function, equation 3.16 is reduced to:

$$L(n|z_{1:t}) = L(n|z_{1:t-1}) + L(n|z_t) \quad (3.17)$$

where

$$L(n) = \log \left[\frac{P(n)}{1 - P(n)} \right] \quad (3.18)$$

Equation 3.17 are constructed based on the pointcloud datastructure. The manipulator workspace over the octomap model are divided into two values; zero for free space, and one for occupied space. Since, Realsense also produce RGB imageries, the map also valid for the occurrence of the color blue, red, and, green:

$$L_{red}(n) = \log \left[\frac{P_{red}(n)}{1 - P_{red}(n)} \right] \quad (3.19)$$

$$L_{green}(n) = \log \left[\frac{P_{green}(n)}{1 - P_{green}(n)} \right] \quad (3.20)$$

$$L_{blue}(n) = \log \left[\frac{P_{blue}(n)}{1 - P_{blue}(n)} \right] \quad (3.21)$$

With logistic encoding, the datastructure of the map is compact, relaxing computational expenses and thus eliminate bottleneck on the rich streaming from the RGB-D sensor. This map model, also known as octomap encodes the posterior of the SLAM solution in equation 1.1. Given the stability of binary values in static environment map-

ping, a dynamic objects introduced in the workspace are not represented by the octomap datastructure.

The SLAM part of this thesis implements the RTAB-Map library and its framework to encode the map model and also the spherical harmonics implementation suggested in the earlier section as the localization engine for the instead the implementation that comes with the libary. The SLAM model starts with the cost function of the map a posteriori:

$$\Phi^* = \underset{\Phi}{\operatorname{argmax}} P(\Phi|c^*, \tau) \quad (3.22)$$

where Φ is the map of the environment from the datastructure infused with the spherical harmonic terms. τ is the the set of waypoints generated by the planner and c^* is the control-space estimate at the configuration c_{ee} .

$$\begin{aligned} c^* &= \underset{c_{ee}}{\operatorname{argmax}} P(C|\Phi, Z) \\ &= \underset{c_{ee}}{\operatorname{argmax}} \frac{P(z|\Phi^*, c_{ee})P(c_{ee}|\Phi^*)}{P(Z)} \\ &= \underset{c_{ee}}{\operatorname{argmax}} \log P(z|\Phi^*, c_{ee}) + \log P(c_{ee}|\Phi^*) \end{aligned} \quad (3.23)$$

here, c_{ee} is the prior estimate of the state of the robot's task space on the map model in 3.22.

With these models, the cost function can be defined as:

$$Cost(c) = \gamma \varepsilon + \sum_{\tau \in C_{cycle}} \Phi^{t-1}[\tau_q z]^2 \quad (3.24)$$

The cost function is differentiated with respect to c to minimized the error ε everytime there is an update on the, sensor reading z , map Φ and command update from V .

To minimize this cost function, equation 3.24 are differentiated. With this cost function, the configuration of the robot in terms of joint angles are discerned.

3.6 SUMMARY

This chapter introduced the mathematical background that will be adopted and implemented into the following chapters. The geometry of r_{mini} configuration space, and its implicit representation, the control space, are explained. Further, this chapter introduce the SLAM solution to estimate the task space of the robot.

CHAPTER 4

EXPERIMENTATION, RESULT AND DISCUSSION

4.1 BENCHMARKING EXPERIMENT DESIGN AND RESULT ON THE SAMPLING-BASED PLANNER IN STATIC ENVIRONMENT

In this research, the planner for the dynamic obstacle avoidance are selected based on the performance of a benchmarking activity. Here, the procedure is explained.

Two poses are set for the benchmark represented in the form of *equation 3.4*. The following vectors explain the numerical value of these poses with respect to the frame attached to the base of *r_mini*.

$$c_{initial} = \begin{bmatrix} 0.0 \\ 0.71 \\ 0.0 \\ 0.71 \\ \dots \\ 0.43 \\ 0.25 \\ 0.42 \end{bmatrix} \quad c_{goal} = \begin{bmatrix} 0.0 \\ 0.71 \\ 0.0 \\ 0.71 \\ \dots \\ 0.46 \\ 0.29 \\ 0.43 \end{bmatrix} \quad (4.1)$$

A box, with dimension, $0.5 \text{ m} \times 0.05 \text{ m} \times 0.575 \text{ m}$, are place in front of the robot, it's pose is described by the vector,

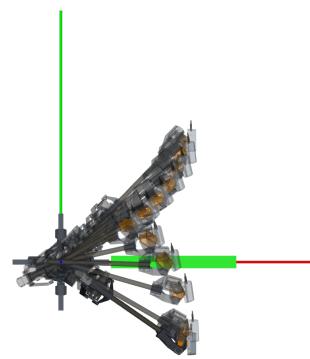
$$c_{box} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ \dots \\ 0.45 \\ 0 \\ 0.2874 \end{bmatrix}$$

Figure 4.1 shows the simulation setup and the planning motion in action. The simulation ran for 50 request from the initial pose to the goal pose. Time processing is given a 10 s limit. The memory limit is set to 1 Mb. The time limit for a request, including the motion and the processing time is set to 3637 s. This thesis uses these configurations and the default configuration of each planners in the MoveIt to start the benchmarking.

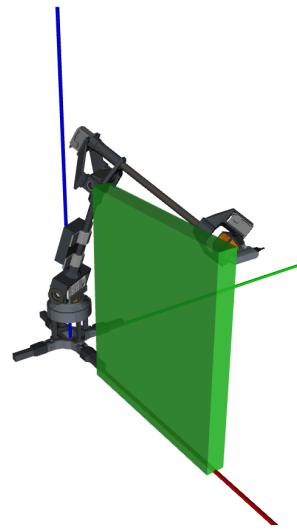
Figure 4.2 shows the compiled statistics of the time the solution were passed to the controller (in this case a virtual controller for simulation of *r_mini* in the simulated environment). RRT requires on average, 0.031 planning time while PRM requires 0.035 planning time from the initial pose to the goal pose when subjected to an obstacle very close to the robot. Wei and Ren (2018) explained the improved RRT algorithms, such as the bi-RRT, and the RRT-connect, solve a query faster. However, based on our benchmarking, vanilla RRT, or base-RRT, and PRM outperform their improved variants when completing the path query between an initial pose and a goal pose. To that end, this research uses vanilla RRT as the scheme for the high-level local planner. This result helps in selecting the motion planner for the dynamic obstacle avoidance.

4.2 EXPERIMENT DESIGN FOR UNPREDICTABLE OBSTACLES USING RRT

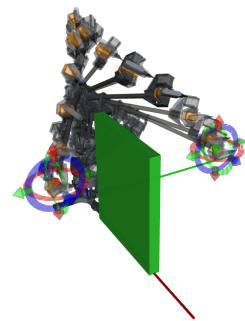
The cyclical space is populated by the RRT-Newton-Raphson and the pipeline where the generated trajectories are then pass to the control pipeline where the controller will



(a)



(b)



(c)

Figure 4.1: The top view of the simulation shown in ,(a) , and the isometric view of the benchmark setup in (b). In (c) r_{mini} attempts to move around the static obstacle placed in it's immediate configuration workspace.

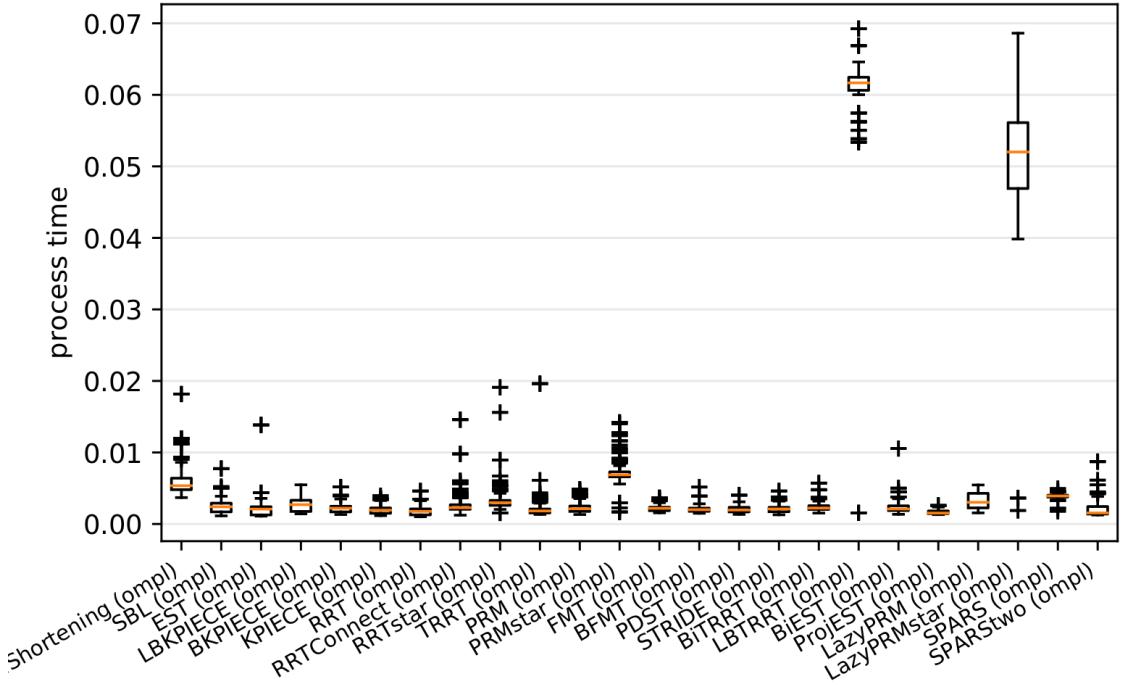


Figure 4.2: The benchmark result when two configurations are defined and pass to the OMPL planner pipeline. All planners completed a 50-cycle query from an initial pose to a goal pose. RRT requires the least amount of processing time at finding the motion planning solution, followed by the PRM.

spline the sparse trajectory waypoints. Two poses are defined in this experimentation which are described by the vectors in *equation 4.2* and the *figure 4.3*

$$c_{initial} = \begin{bmatrix} 0.60 \\ 0.36 \\ -0.60 \\ 0.37 \\ \dots \\ 0.39 \\ 0.04 \\ 0.22 \end{bmatrix} \quad c_{goal} = \begin{bmatrix} -0.18 \\ 0.77 \\ 0.12 \\ 0.59 \\ \dots \\ -0.09 \\ -0.34 \\ 0.34 \end{bmatrix} \quad (4.2)$$

In this experiment, the robot completes a set of ten cyclical space. A static object is introduced between a straight line that connects the initial and the goal pose. The obstacle is introduced in the fourth cycle space. It is then removed at the end of

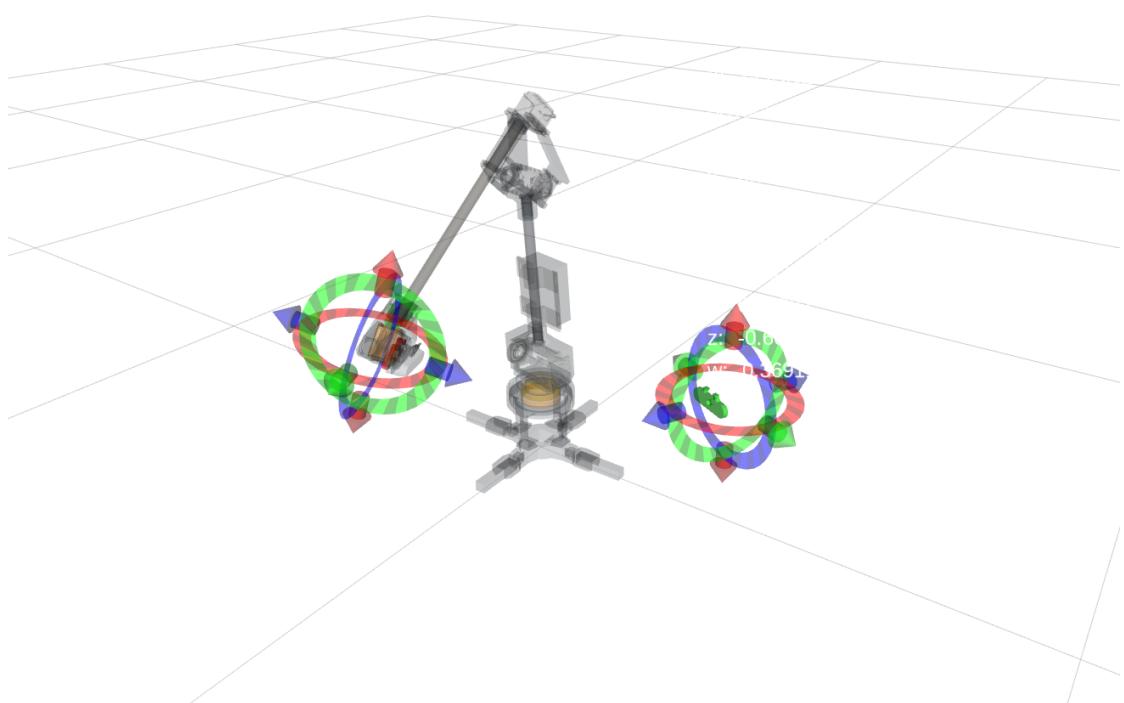


Figure 4.3: The experimentation setup for unpredictable static obstacles where the initial and the goal poses are defined in *equation 4.2*. The green marker shows the initial pose and the red marker shows the goal pose.

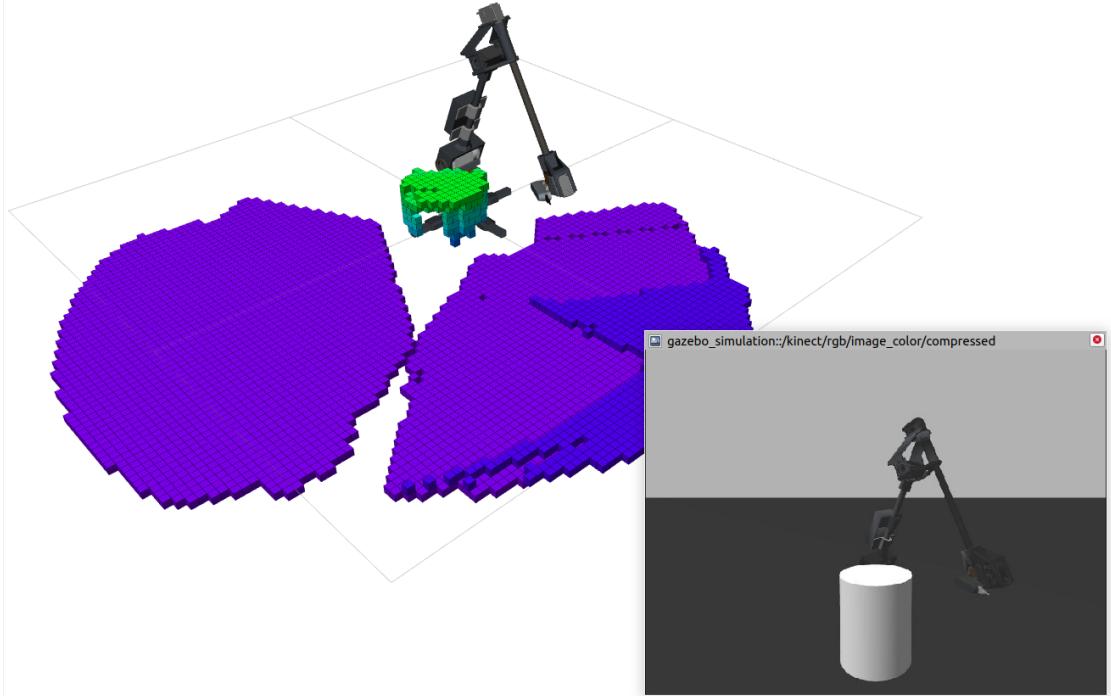


Figure 4.4: The Gazebo Kinect simulation scene is shown in the `gazebo_simulation :: /kinect/rgb/image_color/compressed` window and the feedback representation with octomap map model are shown in the backdrop of the figure. Voxelization coloring is based on the height value. Green voxel clustering shows the representation of the static obstacle in the planning scene. In this diagram, the manipulator is in the fifth cycle space.

the seventh cycle. The obstacle can take any shape, size, and position in the workspace. However, for this experimentation, the obstacle for the simulation is a cylinder with height 0.25 m and radius 0.1 m. The simulation physique engine frontend is the Gazebo software with the ODE library as the engine's backend. In *figure 4.4*, the cylinder is shown via the simulated Kinect sensor feedback, and the representation of the cylinder in the planning scene are represented by the green voxel clustering.

A book with height 0.23 m, length 0.18 m, and thickness 0.02 m was used for the hardware validation.

The simulation was performed and replicated with hardware validation. Five iteration were repeated on both the simulation experimentation and the hardware experimentation.

4.3 EXPERIMENT DESIGN FOR MOVING PLANNING IN DYNAMIC ENVIRONMENT

The cyclical space is populated only by the RRT-Newton-Raphson pipeline. Two poses are defined in this experimentation which has been described in *equation 4.1*. A moving obstacle are placed in front-view of the robot. The obstacle is a cylinder with 0.1 m radius base at 1 m height. The obstacle moves from 0.3 m to 1.7 m away from the robot in oscillation. The period of motion is harmonic, such that, the robot follows a $x = 1 + 0.7 \sin(2\pi(0.0006)\Delta t)$. Two velocities values were used: 50% scale and 10% scale of end-effector's maximum velocity.

The planner is invoked 5 s before the obstacle is placed into the planning scene. The cylinder is directly place into the planning scene such that no motion tracking is necessary for this research. The planner are requested to provide solution for the motion described by the cycle space. Five iterations are done with each given a five minutes runtime. The metric use for this experiment is the time on first collision where, when the prototype touches the cylinder, the iteration is terminated. This experimentatation is done, both, in simulation, and with the real robot hardware. However, for both the simulation and the hardware validation, the obstacle are augmented in simulated environment.

4.4 RESULTS ON PLANNING FOR STATIC AND DYNAMIC OBSTACLE

There are three hardware validations performed. One demonstrate planning when introduced with unpredictable static obstacle. Another regarding the planning under dynamic environment where a synthetic obstacle, moving periodically, cuts through the line of planning in the cyclical space, C_{cycle} . The third experimentation involve the use of RTAB-Map impregnated with the Phase Spherical Harmonics approach to map registration (PHASER) implementation where the state estimation pipeline was swapped with the spherical harmonics approach.

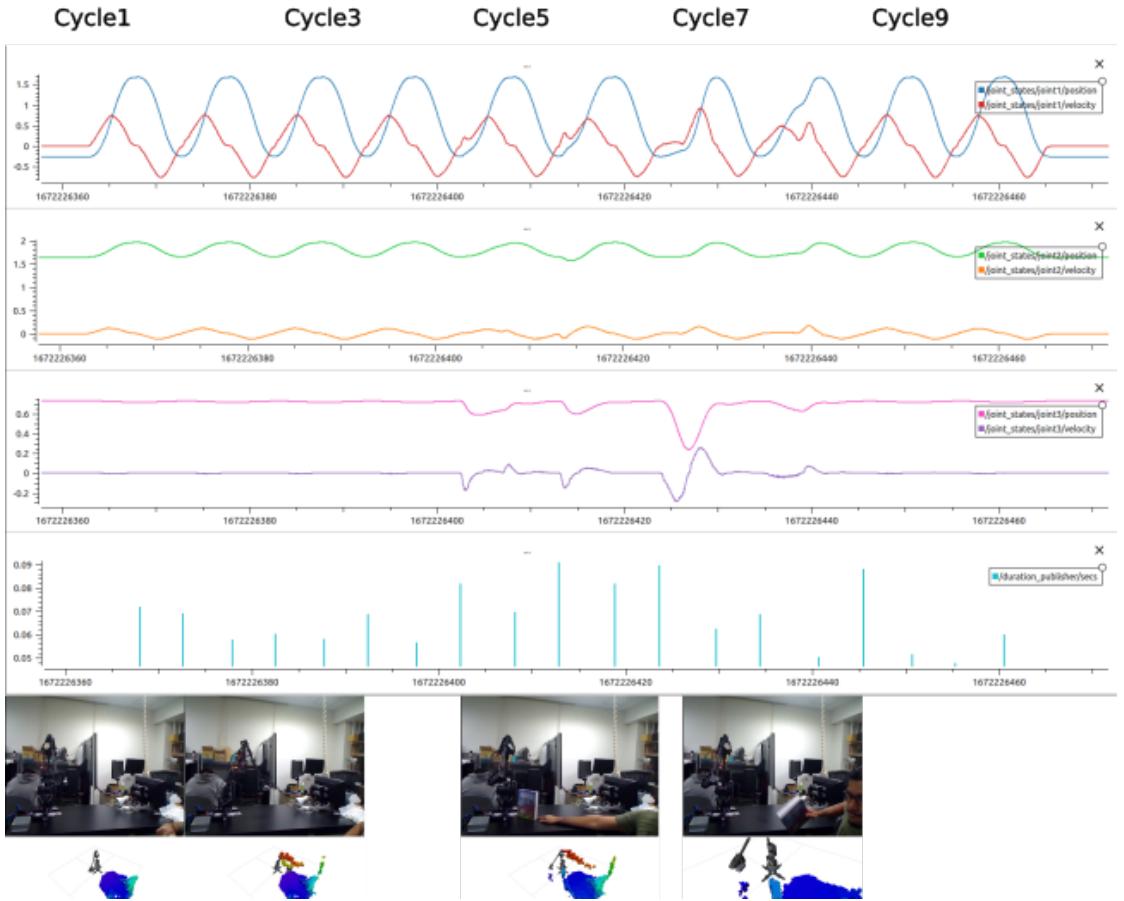


Figure 4.5: The angular displacement and angular velocity for joint1, joint2, joint3, are shown with the planner process time represented by the stick plot. The inset pictures shows the snapshot of the experimentation. The timestamp are represented by the Unix Epoch time format.

4.4.1 Result on Path Planning for Unpredictable Static Environment

The RRT planners avoid the static obstacle reactively under the algorithm 3. Since replanning occurs at the initial and goal pose, collision between the static obstacle and the robot manipulator can only plan to avoid obstacle when the obstacles are placed before a half-cycle space is completed. The obstacle are placed in between the straight line that connects the two poses so that a reactive and successful collision checking, and collision avoidance capability can be demonstrated. When the cycle space generator invoked, r_{mini} avoids the obstacle successfully as depicted in *figure 4.5*.

Table 4.1: The simulated and hardware-connected result of the performance of RRT in a dynamic space. NC stands for No Collision after 5-minute runtime

condition\iteration	time to 1 st collision,(s)				
	1 st	2 nd	3 th	4 th	5 th
simulation _{0.5v}	64	NC	NC	133	18
hardware _{0.1v}	205	16	17	134	13
simulation _{0.5v}	13	23	11	9	13
hardware _{0.1v}	17	4	15	7	10

4.4.2 Result on Path Planning Under Dynamic Environment

Table 4.1, shows the recorded time-to-collision of 20 iterations. The average time to collision is 40 s. There are two iterations with no collision recorded. This poor performance is subjected to the algorithm 3, specifically in line 4 and line 7, RRT is called. Within this call (algorithm 2, line 1 consider an obstacle map that is outdated given the cylinder has moving further towards the manipulator when RRT: line 1. Within the RRT algorithm, there are no mechanism for the robot to stop or move at a lower rate to avoid the cylinder. *Figure 4.6* shows the sequence when the end-effector collide with the cylinder.

Despite the obstacle avoidance fails when the moving cylinder approaches the robot specifically when the centroid of the cylinder is nearing the $x - axes$ of the $c_{initial}$ and c_{goal} , the planner successfully avoid the obstacles when the lines 4 and 7 in algorithm 3 is invoked.

The planner shows reactive behavior when the cyclical space is initialize, via algorithm 3. *Figure 4.7* illustrates such behavior in the simulated environment, and *figure 4.8* shows the same behavior in the hardware reiteration of the experimentation. This is illustrated in *figure 4.9*, where the $joint_2$ and $joint_3$ changes the range of their movement while $joint_3$ changes the rate of its movement.

No significant changes are observed for $joint_4$, $joint_5$ and $joint_6$. This is the implication of the Pieper-condition manipulator design where, none of the $z - axis$ from the first three joints shares the same crossing point, which suggest the rotation acting by

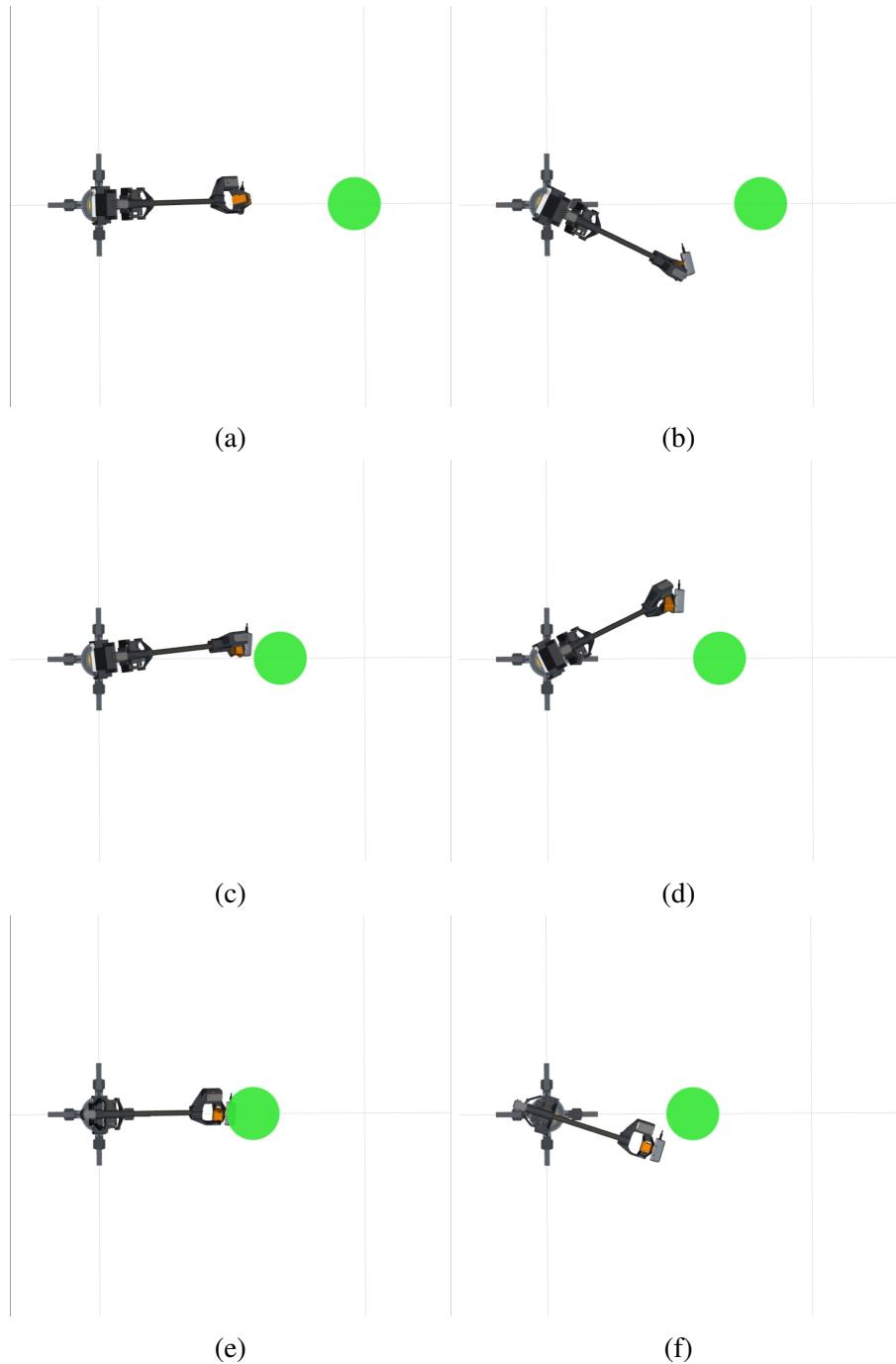


Figure 4.6: This sequence shows the manipulator follows an outdated trajectory and collides with the cylinder despite attempt to move away from the moving cylinder.

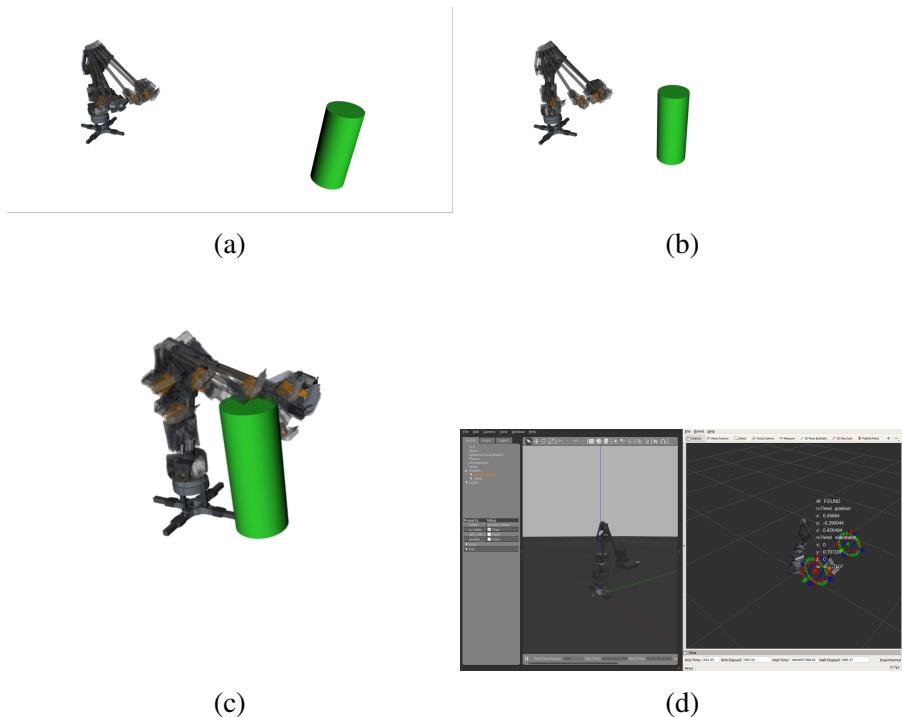


Figure 4.7: The chronology of attempts at avoiding a moving obstacle when the obstacle approaches the robot. The planning algorithm fails at avoiding the cylinder before it passes the $x - location$ of the poses, $c_{initial}$ and c_{goal} . (c) shows the planner successfully provide a non-colliding solution when the cylinder is moving away from robot. (d) shows the Gazebo as the physic engine to replicate the robot hardware and encoders feedback and the cyclical space initialization.

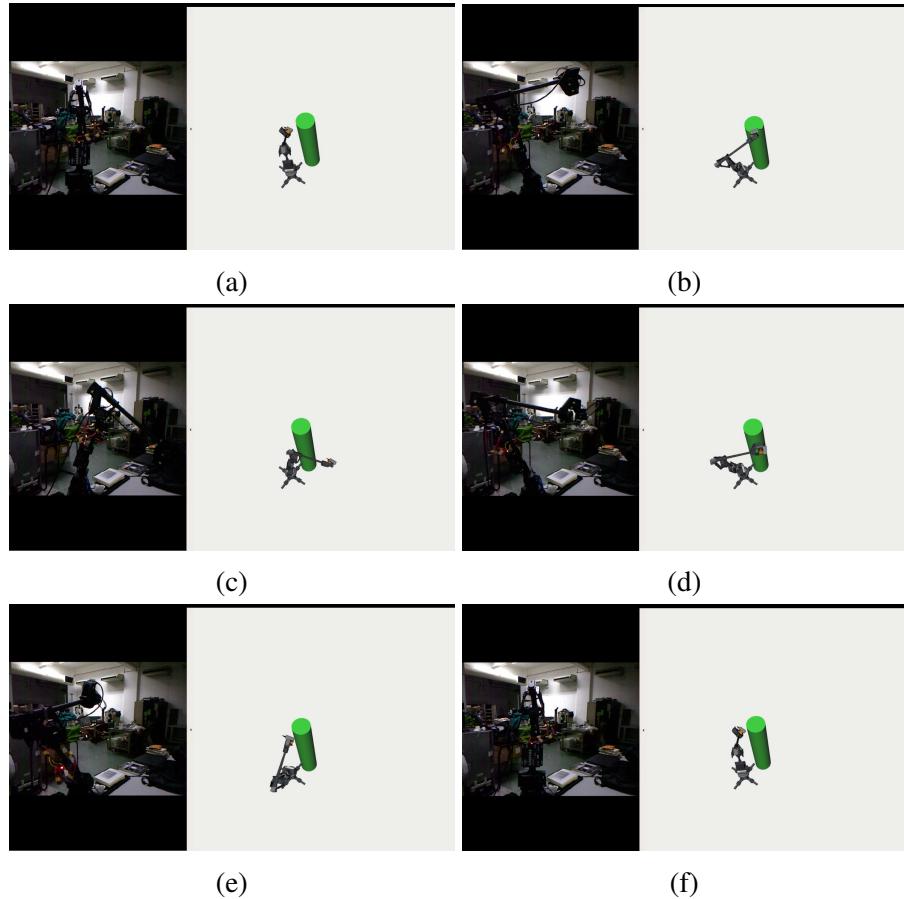


Figure 4.8: The sequence of motion when *r_mini* successfully avoid a moving obstacle when the obstacles at a turning point to move away from the hardware.

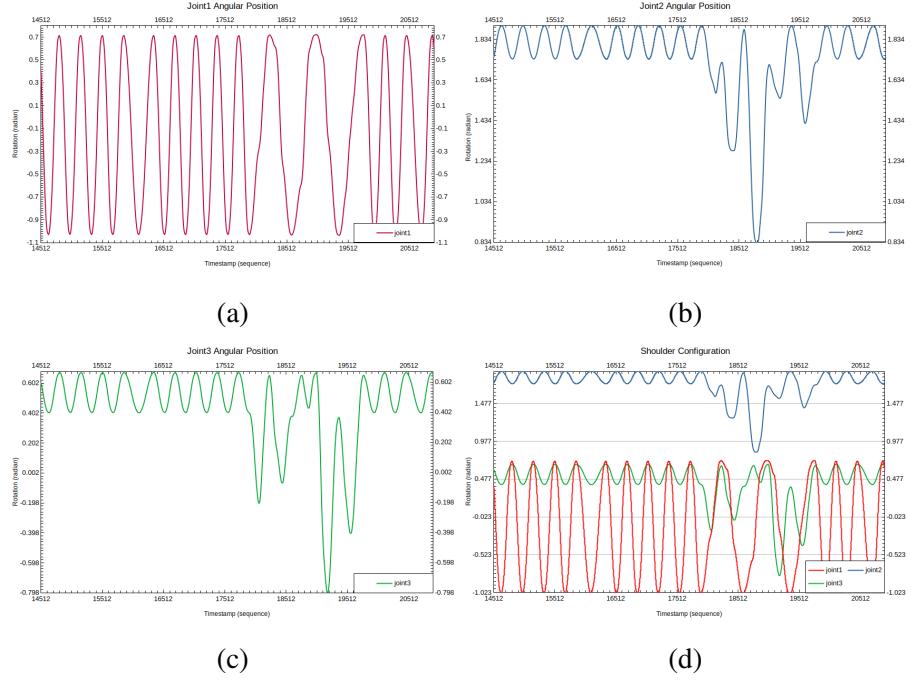


Figure 4.9: Reaction from joint1, joint2, and joint3 shows that the planner together with the cycle space behave reactively towards the moving object. No rapid movement or rate on the last three joints on r_{mini}

these joints are not a linear transformation. Due to the offset (affine transformation) of the joints' axis of rotation, these joints' there is a bijection mapping of these joints to the task-space. Also changes are also observed on the orientation of the frame attached to the end-effector, however, there are no bijection mapping of the three joints to the task-space's orientation.

4.4.3 Result on Obstacle-less Planning with SLAM

The result in *figure 4.10* explains the output of the state odometry pipeline. The \mathbb{R}^3 part of the task space, C_{ee} , is normalized to the origin $(0, 0, 0)$.

As the robot arm moves, changing the pose of the task space, velocity change in the movement together with the angular velocity from the rotation change in the end-effector collapses PHASER's state estimation despite the attempt to use the quaternion and the Hilbert's space to discern the correspondence of two 3D point clouds snapshots. In the figure, there are singularities on most of the process. However, estimations are successfully broadcast between 4.3 s to 6.3 s, 16.1 s to 19.4 s, 32.8 s to 35.2 s, 44.2 s to

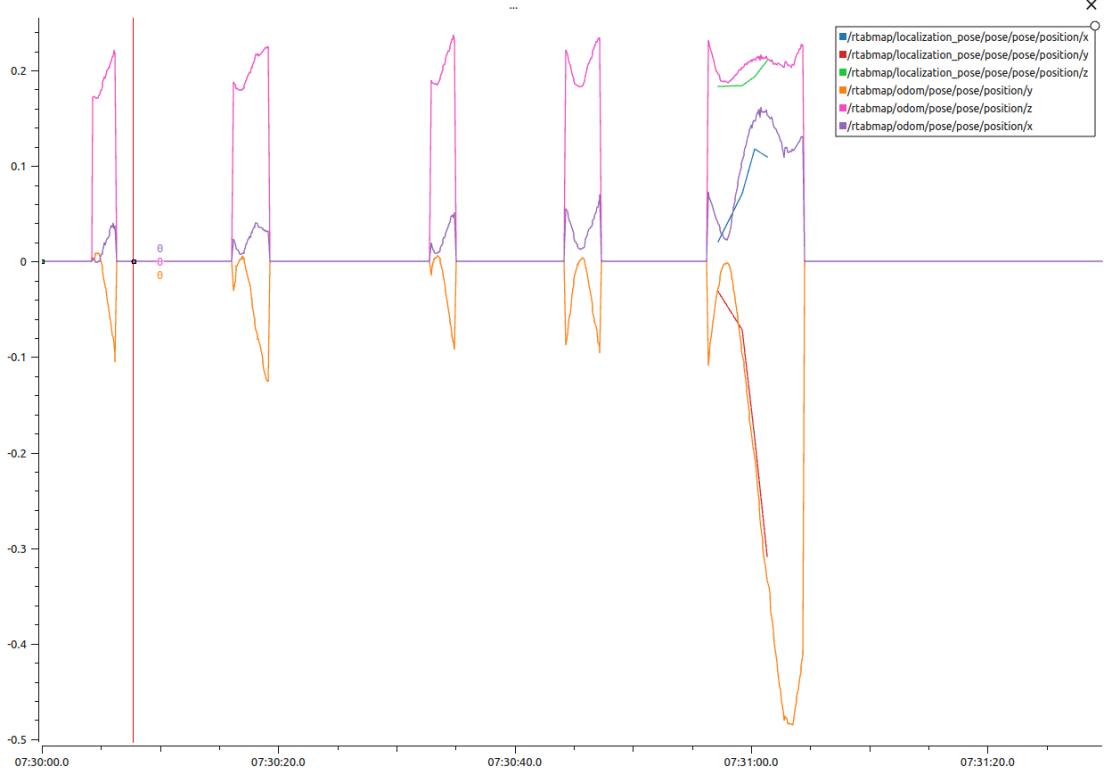


Figure 4.10: The odometry estimate of the end-effector's frame visualize for the \mathbb{R}^3 part of the task space, C_{ee} .

47.3 s, 56.2 to 64.5 s.

The absence of odometry estimation output, where the pipeline is silence, populates 73.4% of the process. The state estimation for the PHASER implementation is intermittent.

The pose estimation from the SLAM pipeline and also the odometry, where the visual odometry are taken into consideration during data fusion, are only appear between 57.2 s and 61.3 s as shown in *figure 4.11*.

In *figure 4.12*, the outputs of the visual odometry and the SLAM estimation are presented by the light blue lines in the left pane of the screenshot. The right shows, the initial and goal pose of the hardware.

The SLAM application on the path planning for this research would not be able to give a stream of pseudo-continuous data output from its pipeline. Since, an intermittent nature of these result were shown from the data, it is observed that, when the state estimation is map into the joint-space configuration, and then feeded into the controller's

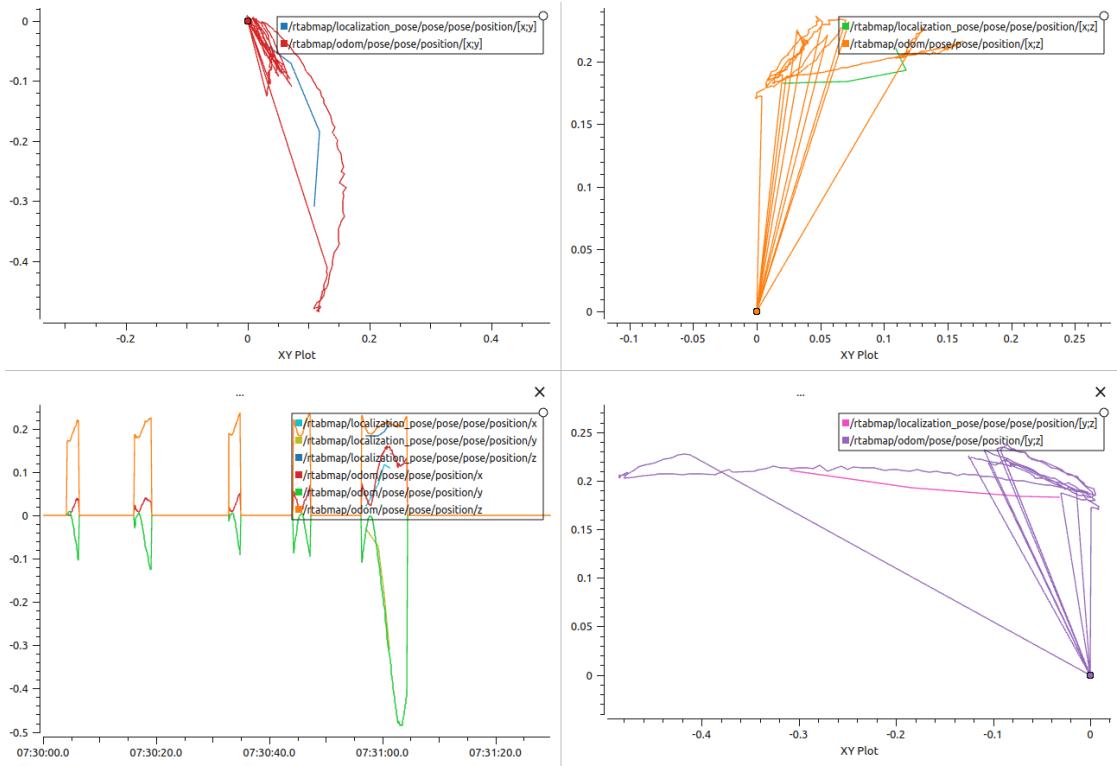


Figure 4.11: The odometry and RTAB-Map state estimation output compared together.

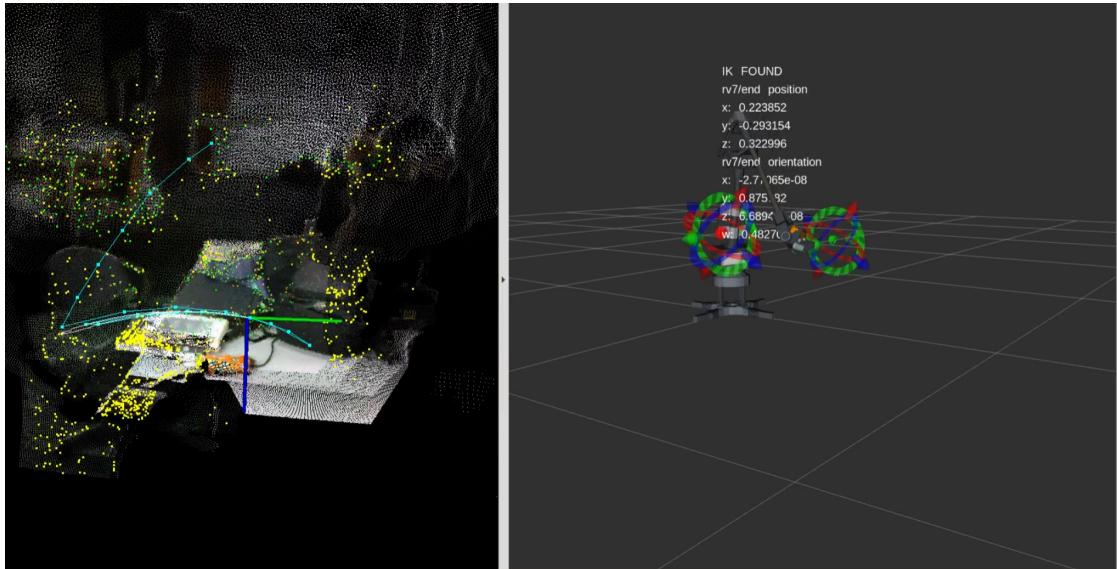


Figure 4.12: This screenshot shows the visual output of the RTAB-Map, where the state estimation and also the visual odometry together with the map of the environment are represented on the left pane of the snapshot. The right pane shows the initial and the goal pose for the C_{cycle}

feedback pipeline, the motor could not abide to the trajectory solution populated by the planner. Hence, currently, with PHASER implementation of the state estimation, the SLAM solution and the planning algorithm will not reconcile and will not work without having an intermediate pipeline that can fill up the gap between the one estimation to the next from the RTAB-Map.

4.5 SUMMARY

In this chapter, the result of four experimentations are presented. These results explain the capability of *r_mini* working in a static environment (via the benchmarking) and in a dynamic environment where this research consequently test on the prototype. The later result shows the feasibility of SLAM at estimating the task space of *r_mini*.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

In this research, a prototype of an industrial robot is developed to investigate the planning and motion control of the manipulator for compliant usage in an industrial setup.

The robot is a 6R manipulator, with six-degree-of-freedom. Each joint is actuated with Dynamixel servo and are back-drivable. The end-effector is equipped with an RGB-D sensor. The robot is named *r_mini*.

Since the Dynamixel motors are not supplemented with a mathematical model, the joints are controlled based on time-parameterized controller where, the set up of each of the motor's velocity profile depends on the angular velocity limits and the angular acceleration limit. The time-parameterized controller was successfully tuned with informed velocity-acceleration limit parameterization. All of the *r_mini* controllers parameters and system configuration, including its driver, are packaged as a stack of ROS packages.

A benchmark was done to ascertain the best sampling-based planner for the *r_mini*'s capability to avoid moving obstacles. The simulation for the benchmark considers a static object, placed in the manipulator's workspace. RRT was selected given its rapid processing time, 0.031 s, at solving a planning objective. Two simulations were done; the first involved introducing a static object with unpredictable shape and placement into the manipulator's line of motion, and the second simulation setup involved a moving object with the shape of a cylinder that was placed and moved in the robot's workspace.

Both the simulations were validated with *r_mini* hardware. The unpredictable

static object invoked reactive motion correction, where no collision were reported. However, when moving obstacles were introduced, the replanning fail to provide collision-free solution. The robot capability to avoid the moving obstacle, although less successful, has not been consistent. The avoidance fails if the RRT is invoke when the obstacle nearly approaching the C_{cycle} .

Although the performance of the RRT on a dynamic obstacle imposed under the cyclical space prescribed in algorithm 3, is not satisfactory, the result shows that the robot is capable at reacting to an obstacles when the obstacle is moving.

This thesis's SLAM implementation, by repurposing RTAB-Map as the SLAM framework and PHASER repurposing as the state estimation pipeline of the RTAB-Map, shows an intermittent and sparse estimation of the C_{ee} which fails to continuously estimate the joint-configuration of the manipulator.

5.2 RECOMMENDATION FOR FUTURE WORKS

This thesis recommends a future work on improved state estimations of the RTAB-Map where the singularities reading during state estimation can be pass to a splining process. The splining would consider the last reading of the RTAB-Map estimation pipeline at, t_{last} , and the output from the *equation 5.1*,

$$\operatorname{argmax}_{t_{last} \leq t, C \in \{C_{cycle}, t\}} \|C - \hat{C}_{last}\|_2 \quad (5.1)$$

where \hat{C}_{last} is the last state estimation of the RTAB-Map before data silence. This approach has the potential of providing pseudo-continuous streaming of feedback for joint controllers; follow-joint-trajectory controller in ROS being the immediate candidate for an encoder-less context. The implication is an encoder-less system that requires no encoders as feedback and, hence, reduces the cost, space of the robot design, and the market price for commercialization of affordable automation system to SME's.

REFERENCES

- Bekris, K. E., & Kavraki, L. E. (2007). Greedy but safe replanning under kinodynamic constraints, In *Proceedings - ieee international conference on robotics and automation*. <https://doi.org/10.1109/ROBOT.2007.363069>
- Biber, P. (2003). The Normal Distributions Transform: A New Approach to Laser Scan Matching, In *Ieee international conference on intelligent robots and systems*. <https://doi.org/10.1109/iros.2003.1249285>
- Burns, B., & Brock, O. (2007). Sampling-based motion planning with sensing uncertainty, In *Proceedings - ieee international conference on robotics and automation*. <https://doi.org/10.1109/ROBOT.2007.363984>
- Cohen, B. J., Chitta, S., & Likhachev, M. (2010). Search-based planning for manipulation with motion primitives. *Proceedings - IEEE International Conference on Robotics and Automation*, 2902–2908. <https://doi.org/10.1109/ROBOT.2010.5509685>
- Corrales, J. a., Candelas, F. a., & Torres, F. (2008). Hybrid tracking of human operators using IMU/UWB data fusion by a Kalman filter. *Proceedings of the 3rd international conference on Human robot interaction - HRI '08*, 193–200. <https://doi.org/10.1145/1349822.1349848>
- Dharmawan, A. G., Foong, S., & Soh, G. S. (2018). Task-constrained optimal motion planning of redundant robots via sequential expanded Lagrangian homotopy. *Journal of Mechanisms and Robotics*, 10(3). <https://doi.org/10.1115/1.4039395>
- Driscoll, J. R., & Healy, D. M. (1994). Computing fourier transforms and convolutions on the 2-sphere. *Advances in Applied Mathematics*, 15(2), 202–250. <https://doi.org/10.1006/aama.1994.1008>
- Du, G., & Zhang, P. (2014). Online serial manipulator calibration based on multisensory process via extended kalman and particle filters. *IEEE Transactions on Industrial Electronics*, 61(12), 6852–6859. <https://doi.org/10.1109/TIE.2014.2314051>
- Du, G., Zhang, P., & Wang, X. (2014). Human-manipulator interface using particle filter. *TheScientificWorldJournal*, 2014, 692165. <https://doi.org/10.1155/2014/692165>
- Faverjon, B. (1984). Obstacle avoidance using an octree in the configuration space of a manipulator. *Proceedings - IEEE International Conference on Robotics and Automation*, 504–512. <https://doi.org/10.1109/ROBOT.1984.1087218>

- Ferguson, D., Kalra, N., & Stentz, A. (2006). Replanning with RRTs, In *Proceedings - ieee international conference on robotics and automation*. <https://doi.org/10.1109/ROBOT.2006.1641879>
- Ferguson, D., & Stentz, A. (2007). Anytime, dynamic planning in high-dimensional search spaces, In *Proceedings - ieee international conference on robotics and automation*. <https://doi.org/10.1109/ROBOT.2007.363166>
- Gasparri, a., Panzieri, S., Pascucci, F., & Ulivi, G. (2006). Pose recovery for a mobile manipulator using a particle filter. *2006 14th Mediterranean Conference on Control and Automation*.
- Haghishipanah, M., Li, Y., Miyasaka, M., & Hannaford, B. (2015). Improving position precision of a servo-controlled elastic cable driven surgical robot using Unscented Kalman Filter, In *Ieee international conference on intelligent robots and systems*. <https://doi.org/10.1109/IROS.2015.7353646>
- Haghishipanah, M., Miyasaka, M., Li, Y., & Hannaford, B. (2016). Unscented Kalman Filter and 3D vision to improve cable driven surgical robot joint angle estimation, In *Proceedings - ieee international conference on robotics and automation*. <https://doi.org/10.1109/ICRA.2016.7487606>
- Hamada, K., & Hori, Y. (1996). Octree-based approach to real-time collision-free path planning for robot manipulator. *International Workshop on Advanced Motion Control, AMC*, 2, 705–710. <https://doi.org/10.1109/AMC.1996.509334>
- Healy, D. M., Rockmore, D. N., Kostelec, P. J., & Moore, S. (2003). FFTs for the 2-Sphere-Improvements and Variations. *Journal of Fourier Analysis and Applications*, 9(4), 341–385. <https://doi.org/10.1007/s00041-003-0018-9>
- Hebert, P., Hudson, N., Ma, J., Howard, T., Fuchs, T., Bajracharya, M., & Burdick, J. (2012). Combined shape, appearance and silhouette for simultaneous manipulator and object tracking, In *Proceedings - ieee international conference on robotics and automation*. <https://doi.org/10.1109/ICRA.2012.6225084>
- Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3), 189–206. <https://doi.org/10.1007/s10514-012-9321-0>
- Hu, J., & Xiong, R. (2017). Contact Force Estimation for Robot Manipulator Using Semi-parametric Model and Disturbance Kalman Filter, 0046(100). <https://doi.org/10.1109/TIE.2017.2748056>
- ISO-8373:2. (2021). *Robotics-Vocabulary* (tech. rep.). International Organization of Standardization.
- Ito, A., Li, J., & Maeda, Y. (2020). SLAM-Integrated Kinematic Calibration Using Checkerboard Patterns, 551–556.

- Jaillet, L., & Siméon, T. (2004). A PRM-based motion planner for dynamically changing environments, In *2004 ieee/rsj international conference on intelligent robots and systems (iros)*. <https://doi.org/10.1109/iros.2004.1389625>
- Janabi-Sharifi, F., & Marey, M. (2010). A Kalman-filter-based method for pose estimation in visual servoing. *IEEE Transactions on Robotics*, 26(5), 939–947. <https://doi.org/10.1109/TRO.2010.2061290>
- Jassemi-Zargani, R., & Neculescu, D. (2002). Extended kalman filter-based sensor fusion for operational space control of a robot arm. *IEEE Transactions on Instrumentation and Measurement*, 51(6), 1279–1282. <https://doi.org/10.1109/TIM.2002.808050>
- Kavraki, L. E., Švestka, P., Latombe, J. C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4), 566–580. <https://doi.org/10.1109/70.508439>
- Klasing, K., Wollherr, D., & Buss, M. (2007). Cell-based probabilistic roadmaps (CPRM) for efficient path planning in large environments. *IEEE International Conference on Advanced Robotics*, 13(February 2016), 1075–1080.
- Klingensmith, M., Sirinivasa, S. S., & Kaess, M. (2016). Articulated Robot Motion for Simultaneous Localization and Mapping (ARM-SLAM). *IEEE Robotics and Automation Letters*, 1(2), 1156–1163. <https://doi.org/10.1109/LRA.2016.2518242>
- Koval, M. C., Dogar, M. R., Pollard, N. S., & Srinivasa, S. S. (2013). Pose estimation for contact manipulation with manifold particle filters. *IEEE International Conference on Intelligent Robots and Systems*, 4541–4548. <https://doi.org/10.1109/IROS.2013.6697009>
- Kruse, E., Gutsche, R., & Wahl, F. (1996). Efficient, iterative, sensor based 3-D map building using rating functions in configuration space. *Robotics and Automation, 1996., (April)*, 1067–1072. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=506850
- Kunz, T., Reiser, U., Stilman, M., & Verl, A. (2010). Real-time path planning for a robot arm in changing environments. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 5906–5911. <https://doi.org/10.1109/IROS.2010.5653275>
- LaValle, S. M. (1998). *Rapidly-Exploring Random Trees: A New Tool for Path Planning* (tech. rep.). [https://www.cs.csustan.edu/\\$%5Csim\\$xliang/Courses/CS4710-21S/Papers/06%20RRT.pdf](https://www.cs.csustan.edu/$%5Csim$xliang/Courses/CS4710-21S/Papers/06%20RRT.pdf)
- Lertpiriyasuwat, V., Berg, M. C., & Buffinton, K. W. (2000). Extended Kalman Filtering Applied to a Two-Axis Robotic Arm with Flexible Links. *The Interna-*

tional Journal of Robotics Research, 19(3), 254–270. <https://doi.org/10.1177/02783640022066851>

Li, J., Ito, A., & Maeda, Y. (2019). A SLAM-Integrated Kinematic Calibration Method for Industrial Manipulators with RGB-D Cameras, In *International conference on control, automation and systems*. <https://doi.org/10.23919/ICCAS47443.2019.8971559>

Lightcap, C. A., & Banks, S. A. (2010). An extended kalman filter for real-time estimation and control of a rigid-link flexible-joint manipulator. *IEEE Transactions on Control Systems Technology*, 18(1), 91–103. <https://doi.org/10.1109/TCST.2009.2014959>

Likhachev, M., Ferguson, D., Gordon, G., Stentz, A., & Thrun, S. (2005). Anytime dynamic a*: An anytime, replanning algorithm, In *Icaps 2005 - proceedings of the 15th international conference on automated planning and scheduling*. www.aaai.org

Liu, K., Sui, J., Yue, N., & Liu, S. (2016). Path planning method of mobile manipulator based on the representation space, In *2016 ieee international conference on mechatronics and automation, ieee icma 2016*, Institute of Electrical; Electronics Engineers Inc. <https://doi.org/10.1109/ICMA.2016.7558582>

Luo, R. C., & Kuo, C. W. (2016). Intelligent seven-DoF robot with dynamic obstacle avoidance and 3-D object recognition for industrial cyber-physical systems in manufacturing automation. *Proceedings of the IEEE*, 104(5), 1102–1113. <https://doi.org/10.1109/JPROC.2015.2508598>

Matuszek, C., Mayton, B., Aimi, R., Deisenroth, M. P., Bo, L., Chu, R., Kung, M., Le Grand, L., Smith, J. R., & Fox, D. (2011). Gambit: An autonomous chess-playing robotic system. *Proceedings - IEEE International Conference on Robotics and Automation*, 4291–4297. <https://doi.org/10.1109/ICRA.2011.5980528>

Meeussen, W., Rutgeerts, J., Gadeyne, K., Bruyninckx, H., & De Schutter, J. (2007). Contact-state segmentation using particle filters for programming by human demonstration in compliant-motion tasks. *IEEE Transactions on Robotics*, 23(2), 218–231. <https://doi.org/10.1109/TRO.2007.892227>

Miseikis, J., Glette, K., Elle, O. J., & Torresen, J. (2017). Multi 3D camera mapping for predictive and reflexive robot manipulator trajectory estimation. *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016*arXiv 1610.03646. <https://doi.org/10.1109/SSCI.2016.7850237>

Mohanan, M. G., & Salgoankar, A. (2018). A survey of robotic motion planning in dynamic environments. North-Holland. <https://doi.org/10.1016/j.robot.2017.10.011>

- Moravec, H. P. (1989). Sensor Fusion in Certainty Grids for Mobile Robots. In *Sensor devices and systems for robotics* (pp. 253–276). https://doi.org/10.1007/978-3-642-74567-6_19
- Nissler, C., Buttner, S., Marton, Z. C., Beckmann, L., & Thomasy, U. (2016). Evaluation and improvement of global pose estimation with multiple AprilTags for industrial manipulators. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2016-Novem.* <https://doi.org/10.1109/ETFA.2016.7733711>
- Osteen, P. R., Owens, J. L., & Kessens, C. C. (2012). Online egomotion estimation of RGB-D sensors using spherical harmonics, In *Proceedings - ieee international conference on robotics and automation*, Institute of Electrical; Electronics Engineers Inc. <https://doi.org/10.1109/ICRA.2012.6225098>
- Otte, M., & Frazzoli, E. (2015). RRTX: Real-time motion planning/replanning for environments with unpredictable obstacles, In *Springer tracts in advanced robotics*, Springer Verlag. https://doi.org/10.1007/978-3-319-16595-0_27
- Paul, G., Webb, S., Liu, D., & Dissanayake, G. (2011). Autonomous robot manipulator-based exploration and mapping system for bridge maintenance. *Robotics and Autonomous Systems*, 59(7-8), 543–554. <https://doi.org/10.1016/j.robot.2011.04.001>
- Payeur, P., Hebert, P., Laurendeau, D., & Gosselin, C. M. (1997). Probabilistic octree modeling of a 3D dynamic environment. *Proceedings of International Conference on Robotics and Automation*, 2(April), 1289–1296. <https://doi.org/10.1109/ROBOT.1997.614315>
- Petrovskaya, A., Khatib, O., Thrun, S., & Ng, A. Y. (2006). Bayesian estimation for autonomous object manipulation based on tactile sensors. *Proceedings - IEEE International Conference on Robotics and Automation*, 2006(May), 707–714. <https://doi.org/10.1109/ROBOT.2006.1641793>
- Pieper, D. L. (1968). *The Kinematics of Manipulators Under Computer Control* (Doctoral dissertation). Stanford University.
- Pilania, V., & Gupta, K. (2015). A hierarchical and adaptive mobile manipulator planner with base pose uncertainty. *Autonomous Robots*, 39(1), 65–85. <https://doi.org/10.1007/s10514-015-9427-2>
- Pires, J. (2007). *INDUSTRIAL ROBOTS PROGRAMMING : INDUSTRIAL ROBOTS PROGRAMMING* (Doctoral dissertation).
- Pomarlan, M., & Sucan, I. A. (2013). Motion planning for manipulators in dynamically changing environments using real-time mapping of free workspace, In *Cinti 2013 - 14th ieee international symposium on computational intelligence and informatics, proceedings*. <https://doi.org/10.1109/CINTI.2013.6705245>

- Rigatos, G. (2009). Particle Filtering for State Estimation in Nonlinear Industrial Systems. *IEEE Transactions on Instrumentation and Measurement*, 58(11), 3885–3900. <https://doi.org/10.1109/TIM.2009.2021212>
- Ruhr, T., Sturm, J., Pangercic, D., Beetz, M., & Cremers, D. (2012). A generalized framework for opening doors and drawers in kitchen environments. *Proceedings - IEEE International Conference on Robotics and Automation*, 3852–3858. <https://doi.org/10.1109/ICRA.2012.6224929>
- Rybski, P., Anderson-Sprecher, P., Huber, D., Niessl, C., & Simmons, R. (2012). Sensor fusion for human safety in industrial workcells. *IEEE International Conference on Intelligent Robots and Systems*, 3612–3619. <https://doi.org/10.1109/IROS.2012.6386034>
- Sawada, Y., Kondo, J., & Watanabe, Y. (2012). UKF-based collision detection and control of parallel-structured two-link flexible manipulators. *International Journal of Innovative Computing, Information and Control*, 8(3 B), 2399–2413.
- Song, K. T., Jiang, S. Y., Wu, C. J., Lin, M. H., Wu, C. H., Chiu, Y. F., Lin, C. H., Lin, C. Y., & Liu, C. H. (2013). Mobile manipulation and visual servoing design of a configurable mobile manipulator. *2013 CACS International Automatic Control Conference, CACS 2013 - Conference Digest*, 239–244. <https://doi.org/10.1109/CACS.2013.6734139>
- Su, J., & Xie, W. (2011). Motion planning and coordination for robot systems based on representation space. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41(1), 248–259. <https://doi.org/10.1109/TSMCB.2010.2051025>
- Sun, P., Chen, J., & Lau, H. Y. (2016). Programming human-like point-to-point approaching movement by demonstrations with Large-Scale Direct Monocular SLAM. *2016 IEEE International Conference on Robotics and Biomimetics, RO-BIO 2016*, 1498–1503. <https://doi.org/10.1109/ROBIO.2016.7866539>
- Ulrich, S. (2011). Extended Kalman filtering for flexible joint space robot control. *American Control Conference*, 1021–1026. <https://doi.org/10.1109/ACC.2011.5990848>
- Um, D., & Ryu, D. (2013). SPAM for a Manipulator by Best Next Move in Unknown Environments. *ISRN Robotics*, 679784(8), 5273–5278. <https://doi.org/10.5402/2013/679784>
- Vannoy, J., & Xiao, J. (2008). Real-time adaptive motion planning (RAMP) of mobile manipulators in dynamic environments with unforeseen changes. *IEEE Transactions on Robotics*, 24(5), 1199–1212. <https://doi.org/10.1109/TRO.2008.2003277>
- Venator, E., Lee, G. S., & Newman, W. (2013). Hardware and software architecture of ABBY: An industrial mobile manipulator. *IEEE International Conference on*

Automation Science and Engineering, 324–329. <https://doi.org/10.1109/CoASE.2013.6653969>

Wang, X., Yang, C., Ju, Z., Ma, H., & Fu, M. (2017). Robot manipulator self-identification for surrounding obstacle detection. *Multimedia Tools and Applications*, 76(5), 6495–6520. <https://doi.org/10.1007/s11042-016-3275-8>

Wei, K., & Ren, B. (2018). A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved RRT algorithm. *Sensors (Switzerland)*, 18(2). <https://doi.org/10.3390/s18020571>