

Master Thesis Report

Multi Robot Motion Planning for Weld Quality Optimization

Poulastya Mukherjee*

B-IT Master Studies in Autonomous Systems

Bonn-Rhein-Sieg University of Applied Sciences

Fraunhofer IPA

Advisors: Prof.Dr. Paul.G.Ploeger**

Prof. Dr. Rainer Bastert††

Julian Diaz Possada‡‡

January 15, 2016

*poulastya.mukherjee@smail.inf.h-brs.de

**Paul.Ploeger@h-brs.de

††rainer.bastert@h-brs.de

‡‡julian.diaz@ipa.fraunhofer.de

Declaration

I, **Poulastya Mukherjee**, hereby declare that this work has not previously been submitted to this or any other university, and that unless otherwise stated, the content is entirely my own work.

Date

Poulastya Mukherjee

Abstract

1 Acknowledgment

Contents

1	Acknowledgment	5
2	Introduction	9
2.1	Motivation	9
2.2	Overview	10
2.2.1	Objective	10
2.2.2	Proposed Contributions	10
2.3	Structure	11
3	Literature Review and State of the Art Analysis	12
3.1	Motion Planning for Industrial Manipulators	12
3.1.1	Motion Planning for multi-DoF manipulators	13
3.1.2	Motion Planning for manipulator and rotary table systems	15
3.2	Robotic Welding Processes	17
3.2.1	Analysis of existing methods of Robtic Welding	17
3.3	Multi Robot Approaches for Welding	19
3.3.1	Centralized and De-centralized planning	19
3.3.2	Master Slave Configuration	20
3.4	Welding as an Optimization Problem	23
3.4.1	Welding Process Parameters Optimization	23
3.4.2	Parameter Optimization of Motion Planning Algorithms	26
3.5	Robotic Welding Architecture and Planning in Abstract Space	28
3.5.1	Program generation for Welding Robots and Simulation of Gener- ated Plans	28
3.5.2	Product, Process and Resource Model	30
3.5.3	Implementation Platform Architecture	31
4	Problem formulation and Task description	32
4.1	Problem formulation	32
4.2	Problem Approach Description	33
5	Multi-Robot Approach for Weld Cost Minimization	34
5.1	Modeling the Rotary Table	34
5.1.1	Architecture of RobotKit	34

5.1.2	Generating 3D Models for Simulation	36
5.1.3	Kinematic Description of the Rotary Table	37
5.1.4	Workpiece Transformation	37
5.1.5	Selected Edge Transformation	37
6	Conclusion	38
6.1	Contributions	38
6.2	Possible Improvements and Future Directions	38

List of Figures

1	Trajectory Algorithm	14
2	Gas Tungsten Arc Welding	18
3	Gas Metal Arc Welding	18
4	Master Slave Coordinate Specification	20
5	Two Robot Coordinate System	22
6	Robot Location Model	25
7	Offline Weld Program Generation	28
8	Product Process Resource(PPR) Architecture	32
9	Exisitng Architecture of RobotKit(cite rk)	35
10	Updated Architecture of RobotKit(cite rk)	35
11	dd	36

2 Introduction

2.1 Motivation

Robotic Welding constitute majority of all robotic applications in industries today [1]. Large scale industries with fixed manufacturing processes - like the car industry - traditionally uses MIG/MAG welding operations in the car body workshops of the assembly lines. Due to falling costs of robots and increasing precision and greater computational prowess, there is an increasing number of smaller businesses, which follow client oriented manufacturing and produce unique products designed specifically for each client- can now afford to use robotic manufacturing processes to deliver highly customizable and precise goods. Welding being one of the most common industrial process has been a subject of lot of ongoing research for finding out the most suitable way to automate the task by using robots[1]. That being said robotic welding is not at all a straightforward task because, it is difficult to parameterize and to ensure weld quality assessment and control[1]. However robotic welding to significant improvement both in terms of improvement in quality of product and reducing manufacturing costs. Along with the significant challenges mentioned above, robotic welding requires significant knowledge both in the domain of the welding process - which skilled workers usually acquire with years of experience - and programming the robot to perform the a welding task even on simple workpieces[2]. A solution prevalent in the industry is the use of a teach pendant to guide the robot's manipulator for each task, in our case a weld seam, and then let it perform the task autonomously[1,2,3]. The main disadvantage of this method is the significant longer teaching time involved with each new workpiece. Thus the need of the hour is a simple programming interface which will include the process knowledge of welding[1] and allow an operator to generate programs for different weld seams without having to reprogram the robot. Even though generating a welding program is quite simple, the complexity of the process increases when several constraints are introduced - maintaining constant velocity during welding, maintaining constant angle between tool tip and weld seam. The welding process can also be affected by collision objects and presence of singular points in the path of the weld tool tip. While avoiding collision is taken care of in the planning level, modern controllers are programmed to handle singularities. However this kind of singularity avoidance usually results in deviation of the behavior of the robot, which can result in poor quality weld surface. An architecture- namely PPR (Product, Process and Robot) - has been proposed in [8] which uses sepa-

rate models for workpiece, robot and welding process and generates motion plans while taking into account the process constraints. Due to the fact it uses sample based planning algorithms, it can easily generate programs in new workspace environments as well as for new 3 workpieces with varying levels of complexity (linear edges, circular edges, discontinuous edges etc).

2.2 Overview

2.2.1 Objective

The main objective of this work is to extend the existing RobotKit[8] software with multi-robot planning capability. A 6 Dof manipulator will be the primary robot while a 2 dof rotary table will fulfill the role of second robot. Quite a number of multi robot approaches involving a manipulator and a rotary table and two manipulators - already exists. Most notable among them are Master/Slave[5] and non-Master/Slave approach[6], centralized[5] and decentralized planning approach[6]. Our work will also evaluate the most suitable method among these based on criterias like performance time and motion cost as mentioned in[1]. The second objective of the work is to improve the time of solution generated and reduce solution cost, or in other words ensure greater compliance with existing manufacturing process model[8]. The third objective of the work is to extend the existing State Validation model to include Singularity and Reachability checking to generate more robust solution path.

2.2.2 Proposed Contributions

1. Extending the robot model of the PPR architecture from single robot welding process to multiple robot welding approach .
2. Extending the state validation model of the PPR architecture to include singularity and reachability check to ensure greater compliance with process model.
3. Improve performance of the planning algorithm by incorporating cost information and replacing the randomized sampler with a sampler that samples based on the cost information.
4. Evaluation of the proposed methods based on the performance changes in time of generating solution and final solution cost.

5. Implementing other types of planning algorithms e.g. Multi query planners such as PRM (Probabilistic Roadmap), SParse Roadmap Spanner algorithm (SPARS), and other Single query planners such as Rapidly-exploring Random Trees (RRT), Expansive Space Trees (EST) and evaluating the performance with respect to the current problem in hand and determining the best suited planning algorithm.
6. Evaluation of performance of the proposed changes of this work compared to existing system based on time required to generate an optimal solution and cost of final solution.

2.3 Structure

- Chapter 2 describes the state of the art work in the fields of robotic path planning, multi robot path planning, robotic welding and optimizing weld planning.
- Section 3 describes the formulated problem and the plausible approach to solve it.
- Section 4 describes the approach taken to solve the initial position optimization of the table.
- Section 5 describes the experimental setup and the results of the experiment.
- Section 6 describes a use case in which the algorithm was implemented and subsequent results visualized.

3 Literature Review and State of the Art Analysis

This chapter discusses the fundamentals of motion planning and related concepts. According to [1] Motion planning consists of the following basic concepts which are prevalent irrespective of the type of planning problem:

- A **state space** definition in which the all probable scenarios related to the planning problem can be expressed
- Modeling **time** respective to the problem, which can determine the speed of the execution of the problem or the order of sequence in which certain actions should be completed.
- **Actions** are generated by plans which manipulate a state. The relationship between an action and state change is generally described by the user.
- **Initial and Goal States** are the defined by the user. Actions are generated in order to convert the initial state to goal state.
- Usually a **criterion** is specified in order to execute the actions to have a desired outcome. The two main goals of specifying a criterion are:
 1. **Feasibility** The goal state can be reached
 2. **Optimality** Optimizes the performance of the planner e.g. reducing the planning time, taking actions with low cost, while achieving the goal state
- **Plan** is generally a sequence of actions which might be simple or complex depending on the above mentioned criteria.

In the following sections we delve deeper into motion planning for a manipulator

3.1 Motion Planning for Industrial Manipulators

Motion planning is one of the most important aspects, when it comes to efficient program generation for industrial robots. Efficient motion planning can not only reduce time of task execution but also increase the rate of task completion by ensuring pitfalls of collision or singular points are avoided. For industrial manipulators generated plans need to be highly accurate and precise, hence large factories, namely automobile manufacturers employ robots which are pre programmed. However when it comes to small

and medium enterprises, where the key goal is to adapt the manufacturing process based on the task at hand, hard coding a robot is not desirable. A very good example of this is welding, where the task changes based on the workpiece to be welded. Workpieces can be of varying complex geometrical structures and the multiple joints may need to be welded at each iteration. Hence to maintain the same level of efficiency and speed of execution it becomes necessary to generate optimal motion plans.

3.1.1 Motion Planning for multi-DoF manipulators

This section will highlight various existing motion planning approaches and discuss the pros and cons of each. In [2](Collision free path planning..) the authors model the state space as a 3D discretized grid in Cartesian coordinate system. The workpiece and fixtures are modeled as collision objects. A heuristic is introduced which determines the distance between goal and current location while the collision avoidance is handled by a function which calculates the distance between current location and nearest collision objects and assigns a cost value based on the distance, as illustrated with equations (1),(2),(3)

$$obs = \sum x_n - m : x_n + m, y_n - m : y_n + m : z_n - m : z_n + m \quad (1)$$

$$d = \sqrt{(x_n - x_c)^2 + (y_n - y_c)^2 + (z_n - z_c)^2} + k * obs \quad (2)$$

$$h = \sqrt{(x_g - x_n)^2 + (y_g - y_n)^2 + (z_g - z_n)^2} + d \quad (3)$$

here k is a constant $[x_c, y_c, z_c]$ defines the current node, $[x_n, y_n, z_n]$ represents the neighboring node and $[x_g, y_g, z_g]$ defines the goal node. The A* algorithm is used to search the tree generated from this definition. Once a path to the goal state has been found, it is checked for inverse kinematic feasibility. The generated joint values are again checked for collision avoidance and then it is passed to the robot program. The main drawback of this approach is as the search tree tries to generate path which are away from obstacles while the workpiece object is designated as a collision object, which results in the manipulator moving far away from the workpiece after it has finished welding one edge. In cases where only one edge needs to be welded there the result will remain unaffected, however for multi edge welding this approach will lead to unnecessary large path costs. Another highlight of this approach was to reduce time of execution by generating trajectories, but the fact that each time the inverse kinematic values are generated; it is rechecked for collision leads to a delay (as collision checking is a computationally expensive process) and eliminates the gains of the trajectory planning. The fact that the

authors did not publish any experimental evaluation of their method in the paper leaves open questions in this regard.

In [3](Online minimum time trajectory planning), the authors propose an algorithm for computing the trajectory of a 6 DoF manipulator which is explained with the help of the following flowchart. The equation (4) for calculating the maximum allowable acceleration and deceleration is

$$\ddot{p}_{max} = \lambda_{min} u_{\ddot{p}} \in \mathbb{R}^m \quad (4)$$

where p is the maximum acceleration in cartesian space.

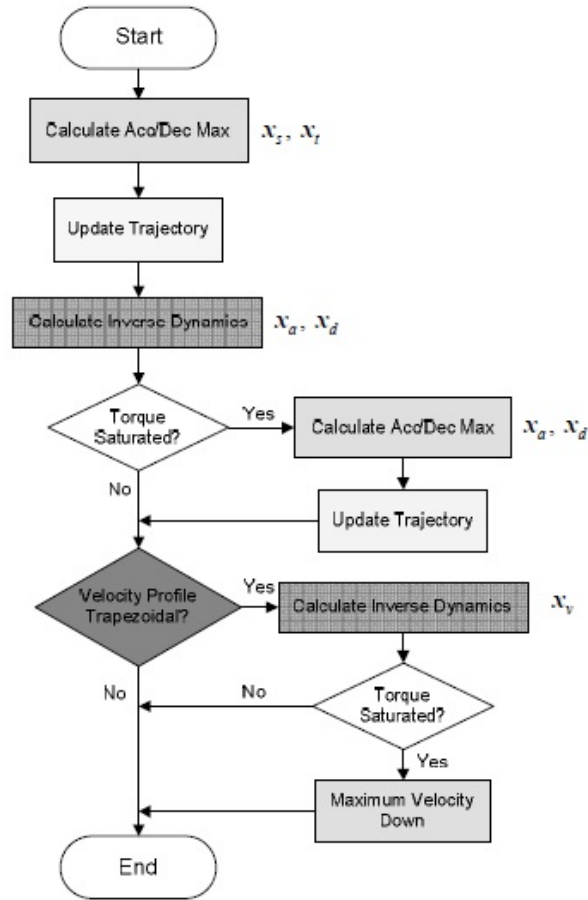


Figure 1: Trajectory Algorithm ⁰

In [4](Integrated Time Optimal Trajectory Planning), the authors propose an algorithm to ensure compliance of the robot motion path and the planned path at high speeds.

The authors demonstrate the working of their algorithm using a parameterized circular path using the equation (5)

$$q = \begin{bmatrix} q_1^*(t) \\ q_2^*(t) \\ q_3^*(t) \end{bmatrix} = \phi(\theta)|_{\theta=\theta^*(t)} = \begin{bmatrix} \phi_1(\theta) \\ \phi_2(\theta) \\ \phi_3(\theta) \end{bmatrix} |_{\theta=\theta^*(t)} \quad (5)$$

In this equation the time parameterization is replaced by $[\theta]$ parameterization which results in a synchronization of joints. Due to this joint velocities can be assigned with the following equation (6) for any velocity profile

$$q = \phi'(\theta)\dot{\theta} \quad (6)$$

After finding the optimal joint velocities, the joint angles are checked whether they satisfy the dynamic constraints. The authors then implemented an Proportional Derivative(PD) control feedback loop as a supplement to the motion planner. The authors propose their own controller apart from the already implemented PD controller and showed that their implementation lead to faster convergence to desired trajectory. The key takeaway from this paper is the parameterized model of trajectory planning which ensures the joint values can be assigned with variables dependent on angle value- $[\theta]$ instead of time.

3.1.2 Motion Planning for manipulator and rotary table systems

A key requirement for producing highest quality weld is the **downhand** welding position, in which the tangent line to the weld is horizontal and the normal vector to the weld is vertical [5](Manufacturing Process Planning for Robotic Arc-Welding Station). Due to this fixed criteria, planned paths for welding often end up having singular points in them, or they encounter collision objects which are difficult to avoid without violating the **downhand** welding position. One of the approaches taken in this regard is to include a till-roll workpiece manipulator or rotary table. In this paper [5] the authors propose a cluster level operation planning with the optimization objective being - minimization of manufacturing time.

The authors divide the welding task into several clusters using a technique called weld seam clustering which arranges the welds into groups that do not require altering positioner configuration. The idea is to find the most optimum path through these clusters. The authors use a generalized TSP (GTSP) algorithm with GI3 heuristics which incorporates the following 3 phases (i) Generalized Initialization, (ii) Generalized Insertion , and (iii) Generalized Improvement. The above mentioned method was able to generate

valid optimized solutions for 70-80% cases with a maximum optimization of 2.6%. A possible reason for this method not generating valid solutions 100% of the time is due to the fact that, generated solutions are not subjected to kinematic constraints and only to the abstract space constraints like collision avoidance and seam selection checking. In [?] (Kinematic and dynamic analysis of 3 dof.) the authors describe a decentralized multi robot planning approach for general industrial tasks like welding, milling etc. with a 6 dof manipulator and a 3 dof rotary table. The dynamic equation is illustrated in (7)

$$M_{OK} = P_i(H_{OK}), H_{OK} = J_{OK}\omega_{i-OK} \quad (7)$$

(7) is applied to every point, where each point shown with k symbol has M_{OK} is the external torque, H_{OK} is the rotational momentum and J_{OK} is the moment of inertia for point k and ω_{i-OK} is the rotational velocity.

3.2 Robotic Welding Processes

In a recent study conducted by the American Welding Association revealed that the current demand for manufacturing welded goods are from small or medium batch manufacturing. In these settings robotic welding provides the most cost effective solution compared to human welders or hard automated setups. In this section we will discuss the various welding methodologies that are compatible with robotic welding and describe them briefly.

In [6](Welding Robots: Technology, System Issues and Application, Pires, J.N. and Loureiro, A. and Bolmsjo, G), the authors state that the most prevalent method of welding in industry is the arc welding method. The two main types of arc welding are gas shielded tungsten arc welding (GTAW) and the gas shielded metal arc welding (GMAW). Apart from these, the various other methods of welding include Laser Beam Welding(LBW),Resistance Spot Welding (RSW), Friction Stir Welding (FSW). Since GTAW and GMAW are the welding methods used for robotic welding in industries, we will elaborate on these.

- *GTAW*: In this process an arc is created between a non-consumable electrode and the work metal which is then shielded by an inert gas to remove any atmospheric contamination. A current of 50-700 A is used to melt and fuse the work metal.
- *GMAW*: In this process a consumable electrode is used. A high current is applied which melts both the electrode and the work metal and fuses them together. An inert gas is used here as well to prevent contamination. The magnitude of current and voltage, size of electrode and type of inert gas used, determines the type of welding method:spray, short-circuiting, globular and pulsed transfer.

3.2.1 Analysis of existing methods of Robtic Welding

Gas Tungsten Arc Welding (GTAW):

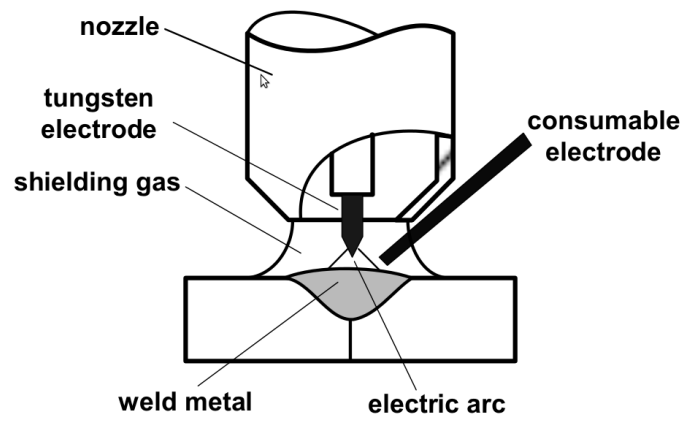


Figure 2: Gas Tungsten Arc Welding⁰

Gas shielded metal arc welding (GMAW):

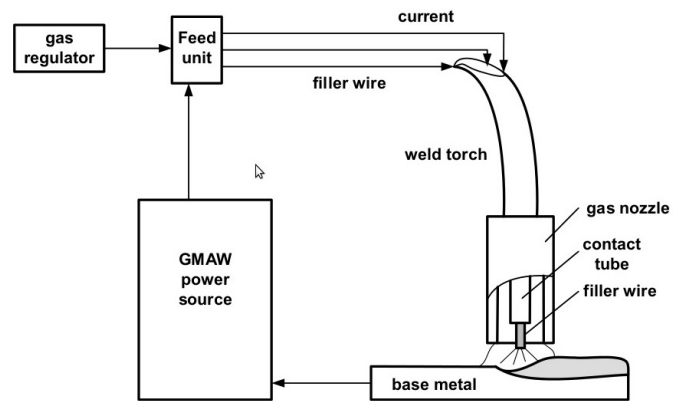


Figure 3: Gas Metal Arc Welding⁰

Table 1: Comparison of GMAW and GTAW welding

	GMAW	GTAW
Power Sources	Constant voltage type	Constant current type
Electrodes	Consumable type	Non-consumable type
Current	Direct current electrode positive applied to weld head	Direct current electrode negative applied to weld head
Welding Speed	A small increase in speed leads to greater weld material penetration but beyond a threshold leads to undercut	Changing welding speed has no effect on quality of weld
Arc Length	between 5-15mm	usually 2-5mm
Electrode Vertex angle	Usually kept at 45deg to normal of the weld edge	Angles between 30deg - 120deg

3.3 Multi Robot Approaches for Welding

One of the major drawbacks of robotic welding is the inability of the welding robot to manipulate the object. While for simple tasks, this does not pose a challenge, however for welding complex work pieces or for avoiding collision in the welding path, it becomes necessary to manipulate the object as well. In this section we will illustrate some of the well known and used approaches for multi robot welding. A well used approach to address this problem, is using two robots instead of one. A 6 dof manipulator is coupled with either a 2 dof rotary table or another 6 dof manipulator. It goes without saying that generating plans for more than one robot is a tedious task all the while maintaining the kinematic and dynamic constraints, as well as following the process parameters. In the next two subsections we will discuss about two popular multi robot planning and control approaches.

3.3.1 Centralized and De-centralized planning

A key point in designing multi robot planning approach is to generate motion plans centrally from either one robot or to use distributed planning with each robot planning its own path and then combining them to form a final plan.

In [7](Comparison of alternate methods for distributed motion planning of robot) the authors discuss a distributed planning approach based on artificial potential framework. The authors consider a case of planning path for three robots. The kinematic and dynamic model is formulated followed by the constraint relationship formulation as illustrated by (8),(9),(10)

$$\dot{q} = v \quad (8)$$

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = E(q)u - A^T \lambda \quad (9)$$

$$M(q) = \begin{bmatrix} [M_A]_{2,2} & 0 & 0 \\ 0 & [M_A]_{2,2} & 0 \\ 0 & 0 & [M_A]_{2,2} \end{bmatrix}, q = \begin{bmatrix} q_A \\ q_B \\ q_C \end{bmatrix} \quad (10)$$

where $q = [x_i y_i]^T \in \mathbb{R}^2$ while the constraint equation is given as follows (11)

$$C(q) = \begin{bmatrix} (x_A - x_B)^2 + (y_A - y_B)^2 - c_{AB}^2 \\ (x_B - x_C)^2 + (y_B - y_C)^2 - c_{BC}^2 \\ (x_C - x_A)^2 + (y_C - y_A)^2 - c_{CA}^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (11)$$

The authors then tested this model with independent parameterization and concluded that the projection based approach is robust but computationally expensive and which increases with each additional step.

3.3.2 Master Slave Configuration

One of the major approaches for multi robot planning has been proposed in [8](kinematic cooperated welding trajectory planning) is master/slave cooperative planning. In this approach an initial kinematic relationship between the frames of the welding seam and the tool tip of the welding robots are established. One of the robots is considered as the master robot and its kinematic chain is defined from the base to its tool tip and for the slave robots the kinematic chain is defined from master robot base to slave robot base to slave robot tool and finally to master robot tool tip. The scheme is illustrated in illustration 4. Master slave configuration can be further divided into tightly coupled mas-

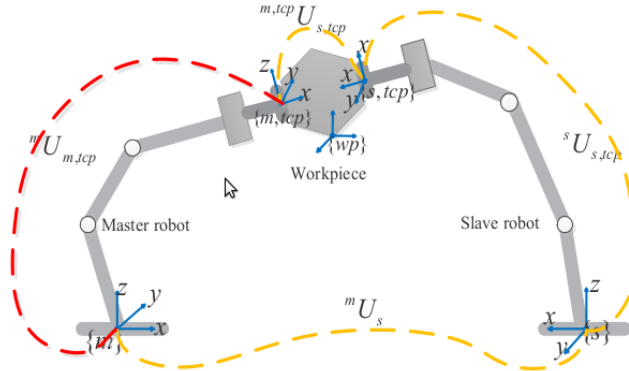


Figure 4: Master Slave Coordinate Specification ⁰

ter/slave mode and loosely coupled master slave/mode. In tightly coupled master slave mode the master and slave robot assignments are fixed and in lightly coupled system the

master slave assignments can be changed. The equation for a general master slave configuration is described in (12), (13) describes a tightly couple master slave configuration and (14) describes the loosely coupled, time based master/slave configuration.

$${}^mU_{m,tcp} = {}^mU_s \cdot {}^sU_{s,tcp} \cdot {}^{s,tcp}U_{m,tcp} \quad (12)$$

$${}^sU_{s,tcp}(t) = {}^mU_s^{-1} \cdot {}^mU_{m,tcp}(t) \cdot {}^{m,tcp}U_{s,tcp} \quad (13)$$

$${}^sU_{s,tcp}(t) = {}^mU_s^{-1} \cdot {}^mU_{m,tcp}(t) \cdot {}^{m,tcp}U_{s,tcp}(t) \quad (14)$$

The authors in [8] then go onto describe three independent scenarios in which master/slave cooperative welding is applied. They are

1. Plate to plate welding - Welding tool tip of the multiple robots should have consistent relative distance.
2. Tube to plate welding - In this mode of operation, the robot which grips the work piece is considered master robot while the welding robot is configured in slave mode.
3. Tube-tube welding - This mode of operation is similar to tube to plate welding.

Based on the results generated from running experiments for the three use cases, it was concluded that the methods described by the authors, achieve the desired goals successfully, however the actual performance of the methods are not reported. A key aspect of master/slave mode of operation that needs to be kept in mind is, for each welding task an accurate description of the seam to be welded needs to be formulated. For simple seam descriptions, this can be easily achieved, however for complex weld shapes the formulation becomes difficult. A counter approach to this is non-master slave approach which is described in [6].

In [6](Offline Kinematics Analysis and Path Planning of Two-Robot Coordination in Exhaust) the authors take two 6 dof robots and establish a coordinated planning based on non-master slave approach. The authors first establish a linked coordinate system for the two robots and the work piece as illustrated in figure 5. In non master/slave mode the pose of an object is set based on time. The mathematical description of the seams to be welded are specified based on which the transformations from robot 1 to work piece edge $R1_{E1}^T$ are calculated and similarly for robot 2 and work piece $R2_{E1}^T$. The seam to be welded is also described mathematically in this approach similar to [8]. The kinematic scheme of the two robots and the work piece is illustrated with the following

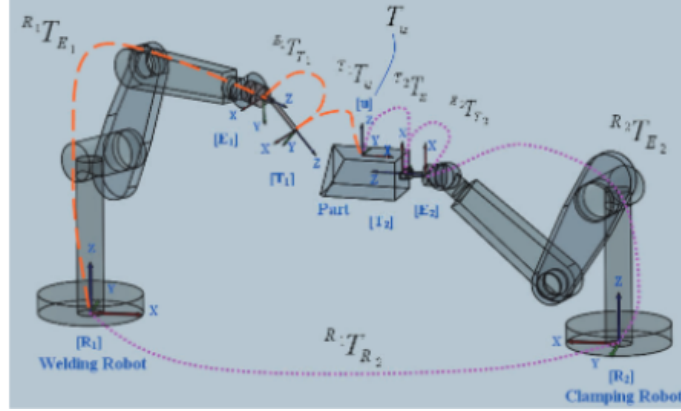


Figure 5: Two Robot Coordinate System ⁰

equation (15).

$${}^{R_1}T_{{}^{E_1}} * {}^{E_1}T_{{}^{T_1}} * {}^{T_1}T_u = {}^{R_1}T_{{}^{R_2}} * {}^{R_2}T_{{}^{E_2}} * {}^{E_2}T_{{}^{T_2}} * {}^{T_2}T_u = {}_uT \quad (15)$$

In (15) ${}^{R_i}T_{{}^{E_i}}$ indicates the transformation matrix of end effector of robot i with respect to base of robot i.

the term ${}^{E_i}T_{{}^{T_i}}$ indicates the transformation matrix of tool tip of robot i with respect to end effector i.

the term ${}^{R_1}T_{{}^{R_2}}$ indicates the transformation matrix of robot 2 with respect to robot 1.

the term ${}^{T_i}T_u$ indicates the transformation matrix of work piece edge u with respect to tool tip i.

The authors conclude in the results section that using this approach will give the user greater capabilities to weld complex edges as the two 6 dof manipulators are more than enough to address collision avoidance and reachability issues. However a drawback of this method to be kept in mind is that for each edge to be welded, (no matter how complex it is geometrically) accurate mathematical description needs to be formulated. This greatly reduces flexibility for welding new edges.

3.4 Welding as an Optimization Problem

Current industrial are mostly pre-programmed and used to perform repetitive tasks. Hence they are unsuitable to be used in small and medium enterprises, where the welding tasks are flexible as the work piece itself may vary from task to task. It goes without saying that the welding process not only needs to be flexible but also needs to adapt based on the shape of the work piece to be welded. Furthermore it needs to finish the welding task in minimum time while consuming minimum resources (power, weld material) as well as ensuring optimal weld quality. This suggests that welding can be treated as an optimization problem with time to weld and motion cost as the cost functions; which are to be minimized. Two major approaches usually taken in this regard are

1. Specify the optimization criteria as extra constraints and generate path plans subject to these additional constraints
2. Adding certain heuristics to existing planning algorithms to generate more optimized plans.

The approaches are elaborated in the following sub sections.

3.4.1 Welding Process Parameters Optimization

In [9](weld quality assessment based on arc sensing), the authors propose a real time analysis of signals in frequency domain to give feedback to the robot controller in real time. The authors record current and voltage consumed during the welding process and then apply power spectral density and time frequency analysis methods to analyze them. The authors use Short Time Fourier Transform(SFTT) to analyze each frequency component of the recorded signals. The authors conducted extensive experiments and came to the conclusion that the developed system performs reasonably well, and is able to optimize the task and take into account any possible defects, however the performance is subject to constraints of accuracy of real time data collection and monitoring. Even though this method is computationally simple and easy to implement, but since the authors only test this method on geometrically simple welding tasks e.g. straight lines, it is difficult to truly gauge the performance of this optimization method. A similar method of regulating the current supplied to the weld tool tip to optimize the weld quality has been presented in [10](Effect of weld parameters on weld quality in sheet to tube..). The authors propose a method in which the current is varied with each point on the weld

trajectory and the tensile strength of the weld spot is measured. From the experiments conducted the authors concluded that as welding progresses the temperature of the weld surface increases and it results in reduced tensile strength of the weld surface. Also continuously increasing the current led to improvement in welding upto a certain level but beyond a certain point the effect of current became detrimental to the weld quality. Even though this paper lacks a model based approach, it provides key insight about the relationship of various parameters to weld quality. In [10](Time optimal trajectory planning for a 6R..) the authors use a genetic algorithm based approach to optimize the trajectory generated for the robot. The authors first model the dynamics of the robot using standard kinematics and inverse kinematic equations. While applying the genetic algorithm, the following steps are taken

1. Fitness function: transforming the constraint equations into fitness functions
2. Determining the generic operators: since the authors use an adaptive genetic algorithm the crossover and mutation parameter change with each step so as to ensure the algorithm neither converges prematurely nor it becomes too wide which leads to non convergence.
3. The initial, final and the middle points are determined through which the robot is expected to travel
4. Apply recursive trajectory planning for each segment
5. Apply adaptive genetic algorithm to compute the fitness function and produce the new generation.
6. Repeat until desired solution with value lesser than desired maximum cost is obtained
7. The final optimal trajectory for each section of the path is obtained along with the time required to traverse it.

The paper lack concrete experimentation and proper results of the experimentation has not been provide which makes it difficult to assess the actual performance of the adaptive genetic algorithm based approach. In [?] (Simulation Planning of Robot Welding) the authors propose a method of process optimization in the following areas of a welding operation; robot location, weld gun selection and motion controlling. For weld gun selection the authors measure the suitability of the weld gun based on the size and shape of the work piece and the work space available as well as the reach-ability of the weld

head in that particular work space. The selection can be made from x type and c type weld heads. Next the authors describe the positioning of the robot manipulator, tool tip, work piece object and other components of the weld cell, in the welding work space. The following diagram 6 describes in brief the hierarchical structure followed in order to define the process. The product and resource position describe the positioning of the

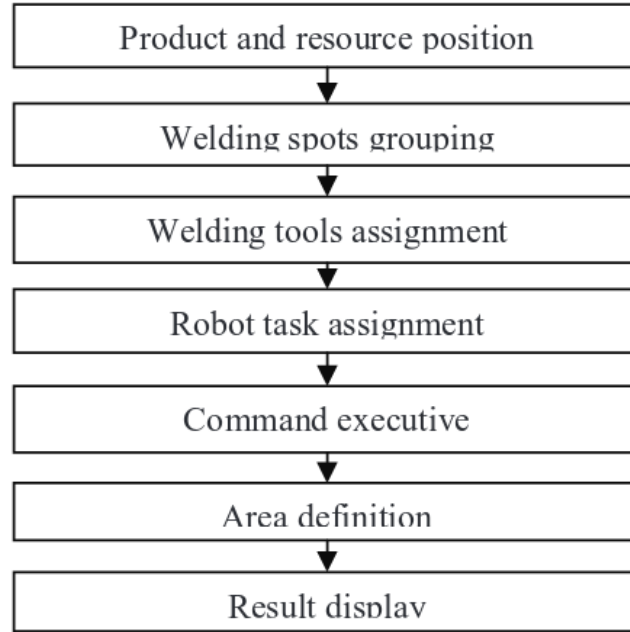


Figure 6: Robot Location Model ⁰

robots, work piece object, tool center point(TCP) and the weld gun. In welding spots grouping, the definition for the seam to be welded are defined. The tool to be assigned based on the work space is determined in welding tool assignment section. After the selection of the weld tool head and the work piece seam, the robot task is assigned. The task section contains definition of the work space in which the plan for the robot is to be generated usually in Cartesian coordinate system. It contains description of the weld seam and any collision objects, if present. After the plan is generated, it is executed on the robot inside the area definition. Finally the results of the planned path is displayed. In the optimization section, the authors break down the final trajectory of the robot in basic motions - such as joint, linear, circle and slew - which can be followed by the robot easily. After this step, the planned path is crash tested and modified accordingly. This paper lays the ground work for a hierarchical welding process model. However a lack of concrete implementation and subsequent testing leaves an open question about the

feasibility of the described model. Few more papers on this topic have been evaluated in the later sections.

3.4.2 Parameter Optimization of Motion Planning Algorithms

In this section we will delve into work that focuses on optimizing the actual planner in order to generate optimal paths. In [11](HGRRT*) the authors propose an optimization based on user given input. The state cost definition is based on potential field framework. The planning algorithm minimizes motion cost by taking path segments with minimum cost while also optimizing the time required to generate the plan. In this approach the authors use a potential field function to guide the sampler of the RRT* algorithm to sample in used desired spaces. The state cost function is calculated using the following equation (??)

$$c(s) = \sum_i \max(-\lambda_i, 0) + \lambda_i e^{\alpha_i d_i(s)^2} \epsilon(0, \sum_i |\lambda_i|] \quad (16)$$

with $\lambda_i \in \mathbb{R}$ and $d_i(s)$ is the distance between work space and robot's reference point. The motion cost for the entire planned path is calculated based on the sequence of states with state costs with the following equation (16)

$$c(P) = \sum_{i=1}^{n-1} c(s_i, s_{i+1}) \quad (17)$$

The proposed algorithm is illustrated as follows

- Sample: Random sample is generated in configuration space with probability $1 - P_{goal}$ and for the goal state the probability is P_{goal}
- Returns the k nearest neighbors of a state q. The value of k depends on dimension of configuration space.
- Returns the path connecting initial state q_{init} and goal state q_{goal} composed by edges of the planning tree.
- Collision free (q1, q2): Checks for collision between states q1 and q2 and returns the result.
- Nearest: the closest node to q is returned.
- Cost: Returns the state cost of q

- EdgeCost(q1, q2): Returns the cost between state q1 and q2

The authors conducted experiments and compared the results with TRRT algorithm. From the results it was concluded that the TRRT takes lesser time to find the solution path, however HGRRT* found shorter and more cost efficient paths.

Experimental optimization of welding parameters is an inefficient and ineffective method of doing so [12]. The authors propose a Gaussian Process Regression Bayesian Optimization Algorithm(GPRBOA), in which the Gaussian Process Regression(GPR) is used to model non parametric complex welding process. Bayesian Optimization Algorithm(BOA) part is applied to balance the modeling and optimization processes. The welding parameters are described as a likelihood function as illustrated in (18)

$$\theta^* = \operatorname{argmax} \log p(y|X, H, \theta) \quad (18)$$

The final model, based on the welding parameters; generated after the GPR process is expressed in (??)

$$x^* = \operatorname{arg max} f(x), x^* \in X \quad (19)$$

In this model, as soon as $f(x)$ is generated, the problem is transformed into an optimization function, with the goal being to find the maximum value of x^* . Due to the reason the exploration and exploitation steps switch between each iteration, this algorithm is able to model the welding process online and optimize it online as well. Compared to other welding parameter optimization methods, this method was able to solve the correct parameter values in 7 iterations. A key point to be noted about this method is that being an online method, it has considerable risks of damaging either the robot or the work piece by drawing in more current, than the recommended amount. It is also difficult to judge from only one experiment whether the algorithm will be able to find out the optimal parameters in few iterations.

3.5 Robotic Welding Architecture and Planning in Abstract Space

One of the major hindrance of using welding robots in industries is in the high cost associated with generation of new programs for welding robots[13](automatic program generation for welding robots..). A human user not only needs to have considerable knowledge of programming to be able to do this, it also requires a considerable time to reprogram a robot. A probable solution is to generate the robot programs offline and simulate the outcome before running the program on the robot. This ensures two things: (i) rectify any anomaly in the generated program contradictory to the desired outcome and (ii) optimize the program to ensure better performance of the robot, be it completing the welding task in lesser time or generating a more robust weld surface. In this section we will discuss the various methodologies and platforms that exists for programming and simulating welding robots offline.

3.5.1 Program generation for Welding Robots and Simulation of Generated Plans

In [13] the authors divide the welding robot offline planning problem in two parts

1. Generating optimized plans for the tool tip within welding constraints. The authors define this as Task space or T-space motion planning.
2. Generating plans between robot home position and start and goal positions of the welding task. This is termed as configuration space planning problem.

The general scheme of operation of the offline program generation is illustrated in the following diagram.

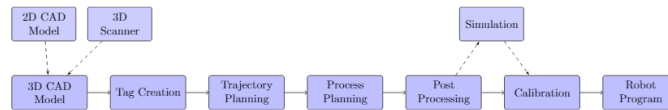


Figure 7: Offline Weld Program Generation ⁰

The actual offline program is generated by the following steps

1. forward and inverse kinematic calculations
2. representing robots, work pieces and weld cell geometrically
3. manipulation and program generation
4. collision detection

Tool Path Planning: The tool path planning space is defined with the following parameter

- t : the position along weld $t \in [0, 1]$
- $\mathcal{R}_{x,y,z}$ rotation around the X,Y,Z axes of tool tip
- cfg : the robot kinematics is represented by $cfg \in 1, 2, \dots, 8$
- rail: position of the robot with respect to linear axis

Motion Planning: The authors use Probabilistic Road Map (PRM) motion planner to generate plans to move the robot from home position and goal positions of the welds. Since the welds are simple paths the authors modified the sampling algorithm of the PRM planner to sample along a guided path without fully populating the tree so that the planner becomes faster. The sampling strategy is described in the following steps.

1. A Gaussian sampling strategy is used where two samples are generated close to the obstacle. If one sample is a collision free sample then it is added to the road-map. This results in sampling is performed close to the obstacles.
2. A number of points were generated close to the start and goal states to ensure they get connected to the roadmap.
3. Some additional points were generated in Cartesian space near weld spots to ensure that a solution is found.

After the two planning stages the authors use a smoothing function to reduce jerky motions generated due to the PRM planner. This optimization resulted not only in smoother paths as well as reduced time for planning. From the results presented by the authors, it is seen that in some cases the planner fails however in most cases the planner is successful but the planning time for each plan varies greatly.

A similar model based approach is found in [14]. The authors generate the CAD models of the work piece, robots and welding cell, which are then used in the simulation platform.

The inverse and forward kinematic of each robot are calculated in a cyclical manner, and separately for each robot. The welding task is represented by a series of way points where each way point contains information about its own geometric position, the torch orientation and the robot's orientation. A search algorithm is then run through the way points to find a connecting path from the start to the goal point. The search algorithm works in the following four steps:

1. a line search is used to find an initial solution
2. if line search fails to find a solution an exhaustive search is launched
3. after a solutions is found the generated path is checked for discontinuities
4. a smoothing algorithm is applied to the generated path to reduce sharp movements of the manipulator

The authors use RRT algorithm to plan path between weld points. The plan is directly generated in the Cartesian Space. A fast distance function from C-Space Toolkit computes the distance based on a distance function in order to generate collision free paths. The authors claimed that this approach has been used to successfully generate plans, however no experimentation results has been provided in the paper to back their claims or establish any performance evaluation.

3.5.2 Product, Process and Resource Model

The solution architecture proposed in [15] is presented as follows:

1. Product model - The product model, consists of the mathematical description of single or multiple-continuous joint seams of a welding work piece, which are selected by the user with a click. The user is able to select the desired edge to be welded from the CAD model.
2. Process model - The process model is a transformation from product to process using Euler transformation. The description is performed in the workpiece frame.
3. Resource model - The resource model contains kinematic and inverse kinematic description of a 6 Dof industrial manipulator.

After the selection of the welding task, the next step is calculation of motion cost, which is performed by the following algorithm 1:

Algorithm 1 Optimal motion cost for collision avoidance for robotic welding

Require: $S_1 = [k_{S1}, \alpha_{S1}]$, $S_2 = [k_{S2}, \alpha_{S2}]$, $T_W^{proc_{opt}}(k)$

Cost calculation for optimal welding angle

```
1: if  $[(Q2AA(T_W^{proc_{opt}}(k_{S1})) > \alpha_{S1}) \vee (Q2AA(T_W^{proc_{opt}}(k_{S2})) > \alpha_{S2})] \wedge$   
    $[(Q2AA(T_W^{proc_{opt}}(k_{S1})) < \alpha_{S1}) \vee (Q2AA(T_W^{proc_{opt}}(k_{S2})) < \alpha_{S2})]$  then  
2:    $RMP_{Opt} = \int_{k_{S1}}^{k_{S2}} Line(k).T_W^{proc_{opt}}(k)dk$   
3: else  
4:    $[k_c, \theta_c] \leftarrow FindIntersection(Line(k), T_W^{proc_{opt}}(k))$   
5:    $RMP_{Opt} = |\int_{k_{S1}}^{k_c} Line(k).T_W^{proc_{opt}}(k)dk| + |\int_{k_c}^{k_{S2}} Line(k).T_W^{proc_{opt}}(k)dk|$   
6: end if  
7: Cost calculation for backward motion  
8: if  $(k_{S2} < k_{S1})$  then  
9:    $RMP_{Backward} = \infty$   
10: else  
11:    $RMP_{Backward} = \emptyset$   
12: end if  
13:  $RMP_{Cost} = RMP_{Opt} + RMP_{Backward}$ 
```

The current state of the art calculates the motion cost based on the angle of rotation from one weld state to another. A state validity checker is also implemented for collision checking and avoidance. Finally RRT* planner is used to solve the defined motion planning problem.

3.5.3 Implementation Platform Architecture

In this section we will briefly highlight the Product Process Resource model's implementation architecture.

The geometrical model of the work piece, the manufacturing process model and the robot model together create the automatic motion planning configuration space. A state validation model runs continuously to check for collisions, while the manufacturing motion cost model calculates the cost. Both of these data are fed to the motion planner, which then generates an optimized motion plan.

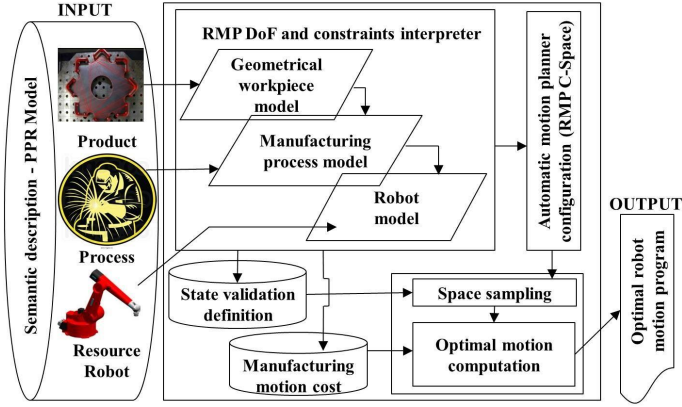


Figure 8: Product Process Resource(PPR) Architecture ⁰

4 Problem formulation and Task description

In this chapter, a list of problems or areas of improvements for [8](PPR) is identified and a model approach based on the literature survey of the previous section is presented. A detailed take on the problems and their respective implementations will be explained in the subsequent chapters.

4.1 Problem formulation

The following problems related to generating successful welding paths were identified:

- **Reachability:** A scenario is highlighting this is presented with the aid of the following diagram(cite diagram). In this scenario, we can see that the selected edge cannot be reached by the 6Dof manipulator. Even though the planner (in this case RRT*) is able to generate a plan successfully, the final solution in the Cartesian space lies outside the reachable workspace of the robot. In order to generate a successful plan, which complies with the optimal definition, it is also necessary for the entire solution to lie in the dexterous work space of the 6 Dof manipulator.
- **Singularity:**
- **Deviation from optimal angle while avoiding collision:** In the following diagram(cite), we can observe that the weld tool tip follows an angle which is not in compliance with the optimal definition in order to avoid the collision. This leads

to poor quality weld.

- **Random Sampling:** The RRT* planner is a fast and efficient algorithm and guarantees a solution asymptotically(cite and rephrase). However as we can see from the diagram below(cite), it generates a lot of random samples which do not at all contribute to finding a proper solution. Hence to make it more efficient, the problem of random sampling can be addressed.

4.2 Problem Approach Description

In this section the proposed approach for the problems described above(cite section) is presented:

- **Incorporating a Rotary Table:** A rotary table can be considered as a 2 Dof which can provide extra degrees of freedom to tackle the problems of reachability and singularity.
- **Extending State Validity Checker:** In order to address the problems of reachability and singularity, it is obvious that we have to incorporate functions to detect reachability and singularity and link them to the state validity checker of the motion planner.
- **Modeling of the Planning Space:** The introduction of the rotary table will create a more complex planning space. Generating a model for the same will help the user to visualize it as well as help to identify the gradient surface, which can be exploited to force the sampler to generate new samples in the areas with lower cost gradient thus ensuring an optimal solution.

5 Multi-Robot Approach for Weld Cost Minimization

In the approach section(cite section) we discussed how introduction of another robot can increase the number of degrees of freedom and hence allow the robot to perform more complex manipulations. Increasing the degree of complexity of motion can allow us to implement better optimization as well as solve the problems of unreachability and singularity. In this chapter we will discuss the integration of a 2 degrees of freedom rotary table into the existing architecture and how that is utilized to optimize the welding task.

5.1 Modeling the Rotary Table

The existing implementation only contained kinematic description of the 6 axis manipulator with the weld tool tip. In the following subsections, we present a model of the 2 degree of freedom rotary table, along with the kinematic description and present the improved architectural model.

5.1.1 Architecture of RobotKit

The following architectural diagram(9) has been color coded for better understanding. Blue represents the core functionalities of RobotKit. Red represents the kinematics and visualization models on which the improvements are implemented. Green represents external configuration files that are loaded inside RobotKit. Yellow represents the OMPL based planning library and pink represents the user interface block.

In the existing architecture the Kernel block creates instances of the kinematics and visualization models of the 6 Dof robot, which update the simulation environment based on the data input from the Information Exchange Pipeline. The planner interface class takes input from the kernel and the information exchange pipeline and user input for welding task. It then creates a planning problem and sends it to the OMPL based planning library. The communication between the planner interface and the planning library takes place using dll files. The project configuration file loads information about the dH parameters of the robot, position of the work piece and other relevant objects and in the weld cell.

In the following architecture diagram (10) the exact blocks which have been updated are highlighted. The kinematic and the visualization models now also includes definition of

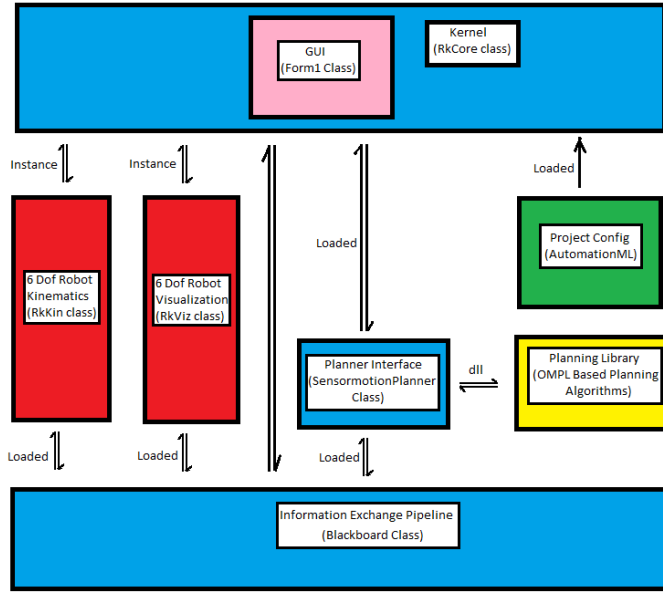


Figure 9: Existing Architecture of RobotKit

rotary table. The project configuration file (highlighted in green) has also been updated to include the dH parameters of the rotary table.

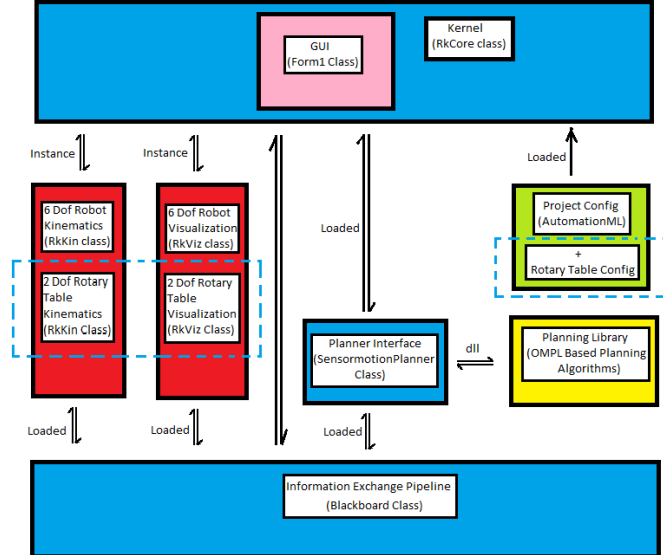
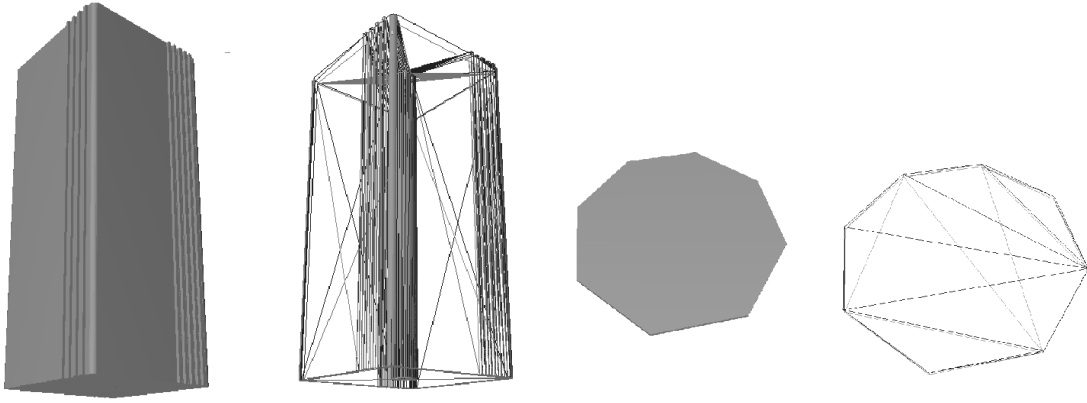


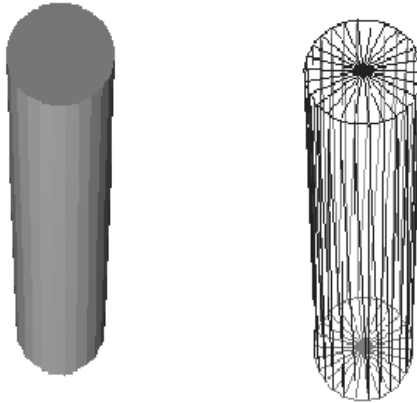
Figure 10: Updated Architecture of RobotKit

5.1.2 Generating 3D Models for Simulation

In order to update the simulation environment, 3D CAD models of the rotary table were created using Solidworks software(cite solidworks), based on the data-sheet specifications. In order to make the simulation mimic the real functioning of the rotary table, the table was divided into 3 separate CAD objects; base, middle joint and top plate. The images for the same are presented herewith.



(a) Base and Top Joint models



(b) Middle Joint model

Figure 11: CAD models of the Rotary Table: Base, Middle and Top

5.1.3 Kinematic Description of the Rotary Table

5.1.4 Workpiece Transformation

5.1.5 Selected Edge Transformation

6 Conclusion

6.1 Contributions

6.2 Possible Improvements and Future Directions

References

- [1] Howie Choset. Coverage For Robotics - A Survey Of Recent Results. *Annals of Mathematics and Artificial Intelligence*, 31:113 – 126, 2001.
- [2] JF Araujo, PB Sujit, and JB Sousa. Multiple UAV Area Decomposition and Coverage. In *2013 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, pages 30–37, Singapore, 2013. IEEE.
- [3] Jie Chen, Wenzhong Zha, Zhihong Peng, and Jian Zhang. Cooperative Area Reconnaissance for Multi-UAV in Dynamic Environment. In *Control Conference (ASCC), 2013 9th Asian*, pages 1–6, Istanbul, 2013. IEEE.
- [4] Alessio Del Bue, M Tamassia, F Signorini, Vittorio Murino, and A Farinelli. Visual Coverage Using Autonomous Mobile Robots for Search and Rescue Applications. In *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–8, Linköping, Sweden, 2013. IEEE.
- [5] Yi Guo and Mohanakrishnan Balakrishnan. Complete Coverage Control for Non-holonomic Mobile Robots in Dynamic Environments. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1704–1709, Orlando, USA, 2006. IEEE.
- [6] Peng Cheng, James Keller, and Vijay Kumar. Time-optimal UAV Trajectory Planning for 3D Urban Structure Coverage. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008.*, pages 2750–2757, Nice, France, 2008. IEEE.
- [7] Hyo-Sang Shin, Cédric Leboucher, and Antonios Tsourdos. Resource Allocation with Cooperative Path Planning for Multiple UAVs. In *UKACC International Conference on Control (CONTROL), 2012*, pages 298–303, Coventry University,UK, 2012. IEEE.
- [8] Yong Bao, Xiaowei Fu, and Xiaoguang Gao. Path Planning for Reconnaissance UAV Based on Particle Swarm Optimization. In *Second International Conference on Computational Intelligence and Natural Computing Proceedings (CINC), 2010*, volume 2, pages 28–32, Wuhan, China, 2010. IEEE.

- [9] Wu Xinggang, Guo Cong, and Li Yibo. Variable Probability Based Bidirectional RRT algorithm for UAV Path Planning. In *Control and Decision Conference (2014 CCDC), The 26th Chinese*, pages 2217–2222, China, 2014. IEEE.
- [10] Wei-Long Yang, Luo Lei, and Jing-Sheng Deng. Optimization and Improvement for Multi-UAV Cooperative Reconnaissance Mission Planning Problem. In *International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2014 11th*, pages 10–15, Chengdu, China, 2014. IEEE.
- [11] Johannes Meyer, Alexander Sendobry, Stefan Kohlbrecher, Uwe Klingauf, and Oskar von Stryk. Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo. In *Simulation, Modeling, and Programming for Autonomous Robots*, pages 400–411. Springer, Dramstadt, 2012.
- [12] Zhu W Li L Teng L Chenglong Y Jiaxun K. Enhanced Sparse A * Search for UAV Path Planning Using Dubins Path Estimation. In *Proceedings of the 33rd Chinese Control Conference*. China, 2014.
- [13] H Cundy and A Rollett. Lissajous’s Figures. *Mathematical Models*, pages 242–244, 1989.
- [14] Steven M. LaValle. Rapidly Exploring Random Trees A New Tool for Path Planning. In *Proceedings of the Mediterranean Conference on Control and Automation*, Sardinia, Italy, 1998.
- [15] Michael Sipser. *Introduction to the Theory of Computation*. Thomson Course Technology, Boston, 2006.