

Master Thesis Report

Multi Robot Trajectory 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

Traditional approach to coverage planning has always focused on developing new techniques for area decomposition and applying search based techniques to navigate an Unmanned Aerial Vehicle (UAV) through the decomposed areas. A key feature ignored in these approaches is the nature of the motion followed by the UAV. Usually the actual motion path of an UAV, using these traditional techniques, is characterized by sharp turns and quite a few start-stop motions. This results in higher time requirement for completing coverage and depletion of excess battery power (as more work needs to be done to move the UAV). Considering the fact that UAVs have limited computational capability and power supply, these two factors are equally important as the percentage of area covered.

In this work we propose algorithms based on geometric functions which have smoother trajectory compared to the lawnmower approach. By conducting experiments using our proposed methods, we were able to prove that algorithms with smoother trajectories take 22.22% lesser time and consume 30% lesser energy compared to lawnmower approach - while achieving atleast 98% coverage.

1 Acknowledgment

I would like to express gratitude to both the supervisors, Prof. Paul G. Ploeger and Dr. Marek Schikora for their continuous guidance and invaluable inputs remarks and engagement through the learning process of this Research and Development project. Furthermore I would like to thank the entire team of Sensor Data Fusion (SDF) from Fraunhofer FKIE for their continued support and the help they provided whenever I faced any difficulty. Last but not the least I would like to thank all my dear friends in Masters in Autonomous Systems (MAS) for supporting me throughout the course of this project.

Contents

1	Acknowledgment	4
2	Introduction	7
2.1	Motivation	7
2.2	Objective	8
2.3	Structure	8
3	Literature Review and State of the Art Analysis	10
3.1	Motion Planning for Industrial Manipulators	10
3.1.1	Motion Planning for multi-DoF manipulators	11
3.1.2	Motion Planning for manipulator and rotary table systems	13
3.2	Robotic Welding Processes	13
3.2.1	Analysis of existing methods of Robtic Welding	14
3.3	Multi Robot Approaches for Welding	15
3.3.1	Centralized and De-centralized planning	16
3.3.2	Master Slave Configuration	16
3.4	Welding as an Optimization Problem	19
3.4.1	Welding Process Parameters Optimization	19
3.4.2	Parameter Optimization of Motion Planning Algorithms	21
3.5	Robotic Welding Architecture and Planning in Abstract Space	23
3.5.1	Product, Process and Resource Model	23
3.5.2	Implementation Platform Architecture	23
4	Problem formulation and Task description	24
4.1	Problem formulation	24
4.2	Task Description	24
5	Conclusion	26
5.1	Contributions	26
5.2	Possible Improvements and Future Directions	26

List of Figures

1	Trajectory Algorithm	12
2	Gas Tungsten Arc Welding	14
3	Gas Metal Arc Welding	15
4	Master Slave Coordinate Specification	17
5	Two Robot Coordinate System	18
6	Robot Location Model	21

2 Introduction

2.1 Motivation

The late 20th century marked the beginning of a shift from electronics age to information age. Be it health-care, entertainment, defense, industries, having thorough information about every aspect of operation is considered critical. Having requisite amount of information not only helps in devising detailed plans but also provides insights about how to deal with a situation in which a plan has failed.

For situations like post disaster damage mitigation, formulating defense reconnaissance plans, or conducting geological survey of a terrain, gathering information is the most important aspect. A common solution for all these three situations is to cover a targeted area with a sensor's footprint so as to gather information about them. The technical term for solution to these classes of problems is called area coverage [1].

Area coverage is usually done by attaching a sensor(e.g. camera, laser scanner) to a robot and moving the robot in the target area in such a way, that information about the entire area is captured by the robot's sensor. Like most robotic solutions, ground robots have been traditionally used for area coverage as well because:

- they are easier to control
- they are robust
- they can carry higher loads compared to aerial vehicles which means they can carry better battery, greater array of sensors, better computers etc.

However certain disadvantages of using ground robots in area coverage scenario are:

- they are terrain dependent (although all terrain robots exists, they are slow and might fail if the terrain is extremely uneven)
- they might not be able to overcome/bypass all obstacles

Herein lies the key advantage of Unmanned Aerial Vehicles or UAVs. Due to the fact that UAVs have an extra degree of motion, they can avoid obstacles much easier compared to ground robots. For disaster mitigation scenarios and defense reconnaissance missions, only gathering information is not sufficient, gathering information in the shortest possible time is also a key requirement. Being airborne, enables the UAVs to take the shortest distance to a goal - which might not be always possible for a ground robot - thus saving a lot of time.

Recent advancements in electronics have led to significant increase in performance of computers while both their size and weight have drastically fallen. This has led to the development of UAVs with better on-board computers and sensors, which have increased their scope of usage especially in area coverage problems. However despite the technological advancements, UAVs still

- have lower computational ability
- and poor batteries

compared to even basic ground robots. Poor batteries directly translate to the fact that UAVs cannot fly for a long time at a stretch. While limited computational abilities restricts it from running advanced algorithms (e.g. search algorithms) on its on-board computer. The combination of these two factors pose a serious challenge to UAVs being used for area coverage problems, which require a robot to move over a large area for long duration of time while performing tasks like recording data and avoiding obstacles. Thus developing efficient techniques for doing area coverage using UAVs are a need of the hour. A possible solution to the problem mentioned above as proposed in [1] is, using multiple UAVs to do area coverage. It has the following advantages,

- reduction in the time required for each UAV to cover an area as now each UAV will have to cover a smaller area
- faster completion of task.

In this work we propose alternate algorithms, which generate a smooth motion trajectory like spirals, Lissajous Curves, hybrid approach. We also use two UAVs for implementing our proposed methods, which will ensure reduced time requirement to achieve goal while not making the process of controlling the overall multi-UAV, system too complex.

2.2 Objective

The aim of this work is to propose alternate methods - spirals, Lissajous Curves, a hybrid approach- to the lawnmower approach [1] and randomized approaches [2] and experimentally determine whether the proposed methods are better or not. For implementing our proposed methods we will use Robot Operating System(ROS) [3]. For simulating and visualizing our results we will use Gazebo [4]

2.3 Structure

- Section 2 describes the formulated problem and the task description.

- Section 3 describes the state of the art work in the field of area coverage using single UAVs, multiple cooperative UAVs and their deployment using libraries and existing simulation platforms.
- Section 4 describes the approach taken to solve the formulated problem.
- 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 [5](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 [6](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.

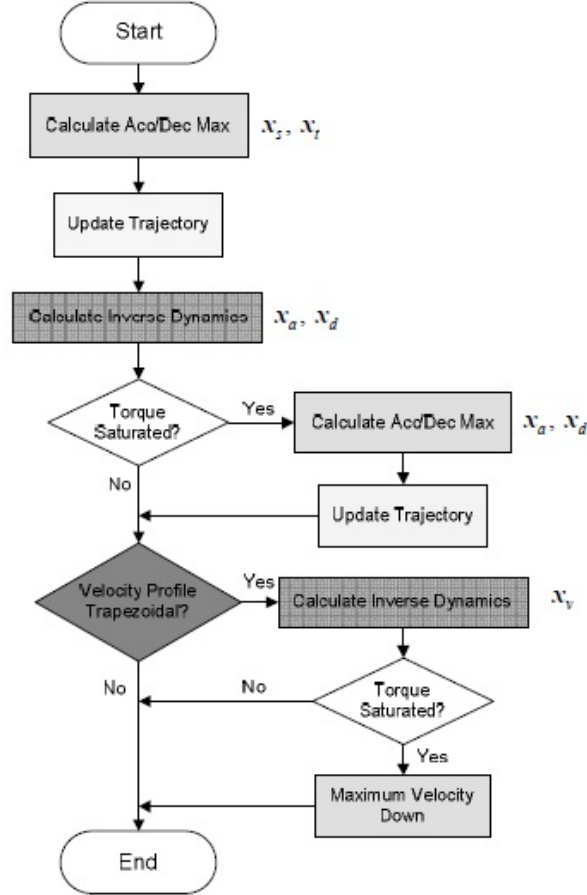


Figure 1: Trajectory Algorithm ⁰

In [7](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 [8](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 [8] 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.

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 man-

ufacturing. 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 [9](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):

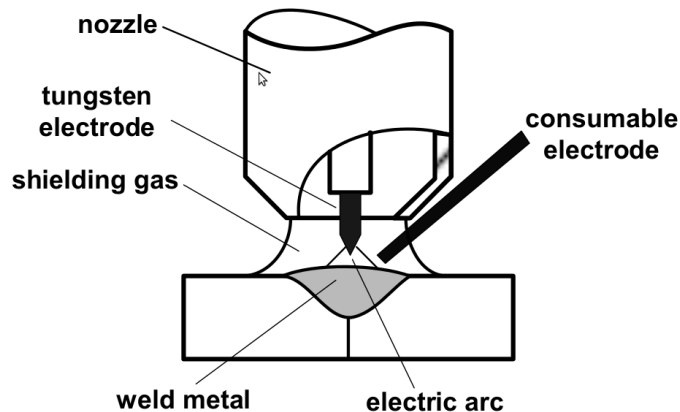


Figure 2: Gas Tungsten Arc Welding⁰

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

Gas shielded metal arc welding (GMAW):

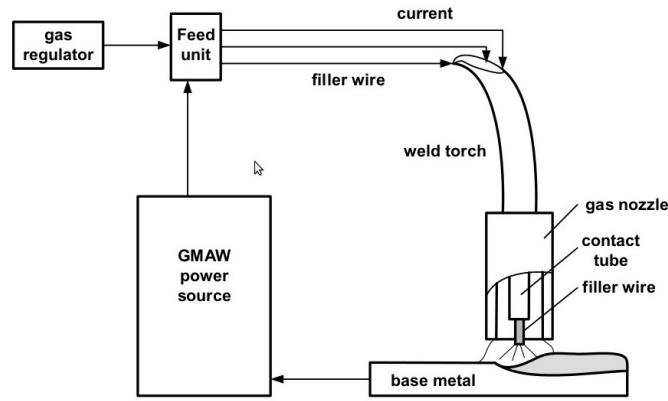


Figure 3: Gas Metal Arc Welding ⁰

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 [10](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 (11),(12),(13)

$$\dot{q} = v \quad (7)$$

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = E(q)u - A^T \lambda \quad (8)$$

$$M(q) = \begin{bmatrix} [M_A]_{2,2} & 0 & 0 \\ 0 & [M_A]_{2,2} & 0 \\ 0 & 0 & [M_A]_{2,2} \end{bmatrix}, q = \begin{bmatrix} q_A \\ q_B \\ q_C \end{bmatrix} \quad (9)$$

where $q = [x_i y_i]^T \in \mathbb{R}^2$ while the constraint equation is given as follows (14)

$$C(q) = \begin{bmatrix} (x_A - x_B)^2 + (y_A - y_B)^2 - c_{AB}^2 \\ (x_B - x_C)^2 + (y_B - y_C)^2 - c_{BC}^2 \\ (x_C - x_A)^2 + (y_C - y_A)^2 - c_{CA}^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (10)$$

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 [11](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 master/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 (11), (12) describes a tightly couple master slave configuration and (13)

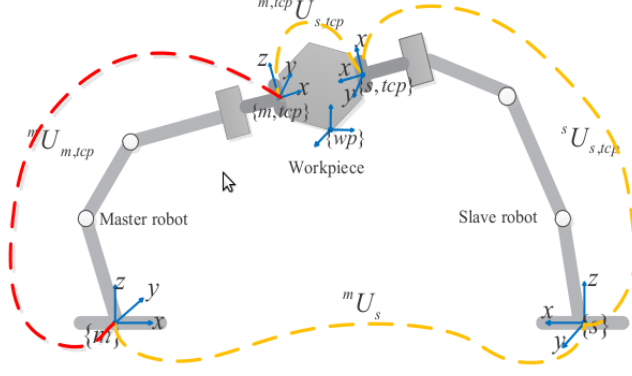


Figure 4: Master Slave Coordinate Specification ⁰

describes the loosely coupled, time based master/slave configuration.

$${}^mU_{m,tcp} = {}^mU_s \cdot {}^sU_{s,tcp} \cdot {}^{s,tcp}U_{m,tcp} \quad (11)$$

$${}^sU_{s,tcp}(t) = {}^mU_s^{-1} \cdot {}^mU_{m,tcp}(t) \cdot {}^{m,tcp}U_{s,tcp} \quad (12)$$

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

The authors in [11] 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 [9].

In [9](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

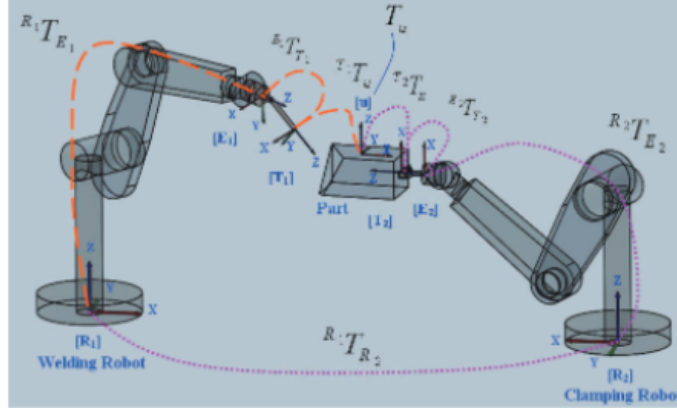


Figure 5: Two Robot Coordinate System ⁰

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 [11]. The kinematic scheme of the two robots and the work piece is illustrated with the following equation (13).

$$R_1T_{E_1} * E_1T_{T_1} * T_1T_u = R_1T_{R_2} * R_2T_{E_2} * E_2T_{T_2} * T_2T_u = T_u T \quad (14)$$

In (14) $R_iT_{E_i}$ indicates the transformation matrix of end effector of robot i with respect to base of robot i.

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

the term $R_1T_{R_2}$ indicates the transformation matrix of robot 2 with respect to robot 1.

the term T_iT_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. 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 [12](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(STFT) 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 [2](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 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 [2](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 robots,

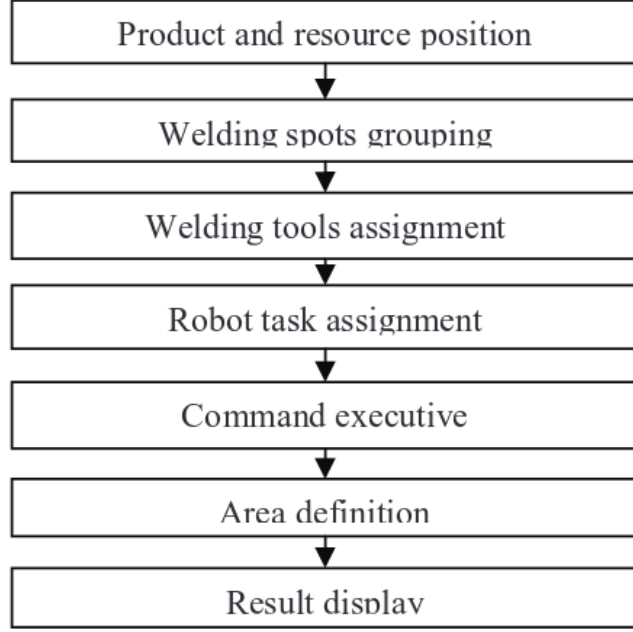


Figure 6: Robot Location Model ⁰

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 [13](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 (15)

$$c(s) = \sum_i \max(-\lambda_i, 0) + \lambda_i e^{\alpha_i d_i(s)^2} \epsilon(0, \sum_i |\lambda_i|) \quad (15)$$

with $\lambda_i \in \Re$ 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)

$$c(P) = \sum_{i=1}^{n-1} c(s_i, s_{i+1}) \quad (16)$$

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.

3.5 Robotic Welding Architecture and Planning in Abstract Space

3.5.1 Product, Process and Resource Model

3.5.2 Implementation Platform Architecture

jjdjdd

4 Problem formulation and Task description

4.1 Problem formulation

Recent works in the field of area coverage using UAVs, is focused on two key aspects for improving area coverage algorithms

- efficient methods of area decomposition [5, 7, 9]
- application of existing search based algorithms to generate coverage paths [11, 10, 12, 2]

A key aspect missing from the above mentioned works is that these works do not focus on the actual motion path followed by the UAVs, which in most cases is simple back and forth motion or a lawnmower type approach, as proposed in [1]. Though the proposed methods in [5, 7, 9, 11, 10, 12, 2] ensure complete coverage, they do not specify any means to reduce the time of coverage. Reducing the time of area coverage is extremely important, to ensure the UAV is able to complete the area coverage task within the time constraints imposed by its limited power supply. Another problem in lawnmower approach is that, during the course of the motion a UAV has to accelerate and decelerate multiple times ?? which increases the time of coverage and also leads to higher power consumption.

4.2 Task Description

In order to compare Like any standard scientific work, we will carry out a state of the art evaluation of area coverage, focusing on area coverage using UAVs in section: ??. Apart from area coverage using UAVs, we will also focus on multi UAV cooperative systems and deployed libraries and existing platforms.

We will then describe our approach in section: ?? where we will focus on formulating mathematical models for proposed algorithms. The following methods are proposed

- Spiral method
- Lissajous Curves based method
- Hybrid approach

For our implementation we will write a ROS package in C++ language which will contain implementation of the proposed algorithms along with some other key functionality for incorporating multi-agent behavior among the two UAVs. We will also implement

the lawnmower approach and Rapidly Exploring Random Tree(RRT) approach. In the experiment section we will run each of these algorithms while varying various parameters like size and shape of the area to be covered, number of commanded waypoints. We will then formulate a cost function to determine which method performs the best.

5 Conclusion

5.1 Contributions

In this work we proposed a geometry based approach to area coverage problem using UAVs. We proposed that since UAVs have limited computational capabilities, using algorithms which require very little computational power can be helpful in creating better solutions. Based on the experiments carried out we came to the conclusion that for doing coverage of small areas lawnmower approach is more suited as it ensures complete coverage in minimal time, however from table ??, we can clearly conclude that spiral method is the best method overall as it ensures atleast 98 % coverage while having minimum energy cost and time to completion. Along with the spiral method, the hybrid method also shows some promise, but its performance in small areas is really poor. The RRT method does not guarantee complete coverage.

Regarding the fact that a smooth motion of the UAVs during its course of flight is a requirement not only for capturing information uniformly but also ensuring a high level of flight stability, the results published in ?? clearly show that the spiral method gives the best performance as it ensures its acceleration and deceleration are within a certain range thus ensuring greater flight stability

5.2 Possible Improvements and Future Directions

A very important future work is to implement the proposed algorithms on a real system and determine whether their performance is in accordance with the findings of this work. In terms of improving the performance of the algorithm, introducing machine learning to make the UAVs autonomously decide which algorithm to use for which scenario will be a big improvement. The primary requirement for creating such a system would be to run multiple number of test to gather enough data.

Apart from the above mentioned directions, changing the orientation and position of the sensor on the UAV and studying how it affect coverage performance is also a possible extension of this work.

References

- [1] Howie Choset. Coverage For Robotics - A Survey Of Recent Results. *Annals of Mathematics and Artificial Intelligence*, 31:113 – 126, 2001.
- [2] 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.
- [3] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, volume 3, page 5, Kobe, Japan, 2009.
- [4] Nathan Koenig and Andrew Howard. Design and Use Paradigms for Gazebo, an Open-Source Multi-robot Simulator. In *International Conference on Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ*, volume 3, pages 2149–2154, Sendai, Japan, 2004. IEEE.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.

- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.