COLLABORATIVE ROBOTIC MOTION PLANNING BETWEEN MULTIPLE ARMS

An Undergraduate Research Scholars Thesis

by

MARC D. RICCIONE

Submitted to the LAUNCH: Undergraduate Research office at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Faculty Research Advisor: Dr. Shawna Thomas

May 2021

Major: Computer Science

RESEARCH COMPLIANCE CERTIFICATION

Research activities involving the use of human subjects, vertebrate animals, and/or biohazards must be reviewed and approved by the appropriate Texas A&M University regulatory research committee (i.e., IRB, IACUC, IBC) before the activity can commence. This requirement applies to activities conducted at Texas A&M and to activities conducted at non-Texas A&M facilities or institutions. In both cases, students are responsible for working with the relevant Texas A&M research compliance program to ensure and document that all Texas A&M compliance obligations are met before the study begins.

I, Marc Riccione, certify that all research compliance requirements related to this Undergraduate Research Scholars thesis have been addressed with my Research Faculty Advisor prior to the collection of any data used in this final thesis submission.

This project did not require approval from the Texas A&M University Research Compliance & Biosafety office.

TABLE OF CONTENTS

ABSTRACT

Collaborative Robotic Motion Planning Between Mutliple Arms

Marc D. Riccione
Department of Computer Science and Engineering
Texas A&M University

Research Faculty Advisor: Dr.Thomas
Department of Computer Science and Engineering
Texas A&M University

Motion planning simply stated is defined as finding a collision-free path between a start and a goal for a movable object, such as a robot [1]. Motion planning is often applied to interactions between robot arms, or between a human and a robot arm in constrained spaces such as those in a factory. A method to solve these problems is presented by combining the solution to constrained spaces, Reachable Volumes[2] and the solution to multiple interactions, Interaction Templates[3]. This integration is done by replacing the motion planning algorithm that is utilized in Interaction Templates with the algorithm for Reachable Volumes. This integration was demonstrated in a virtual environment that was created specifically for motion planning. Experiments were run on two different methods, Interaction Templates and the proof of concept that integrates Reachable Volumes and Interaction Templates. These demonstrate the effectiveness of the method when compared against the prior Interaction Templates standalone.

ACKNOWLEDGMENTS

Contributors

I would like to thank my faculty advisor, Dr. Thomas, and my graduate student advisor, James Motes, for their guidance and support throughout the course of this research.

Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

Finally, thanks to my family for their encouragement and for their patience and love.

The hardware used for Collaborative Robotic Motion Planning Between Multiple Arms were provided by Dr. Thomas. The software used for Collaborative Robotic Motion Planning Between Multiple Arms were provided by the Parasol Laboratory in the Department of Computer Science and Engineering at Texas A&M University.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

Undergraduate research was supported by the Department of Computer Science and Engineering at Texas A&M University.

1. INTRODUCTION

Motion planning simply stated is defined as finding a collision-free path between a start and a goal for a movable object, such as a robot. Motion planning is applied in a variety of scenarios. One application of motion planning can be seen in Amazon's warehouses. In these warehouses, Amazon's warehouse robots move packages from point A to point B while also dodging each other. Another example of motion planning would be in virtual prototyping. Motion planning is used in virtual prototyping to simulate a process such as a assembly line planning in order to prevent collisions on a assembly line conveyor belt. A third example of motion planning would be in video games. In video games, motion planning is used to move objects or characters without colliding with each other.

Motion planning between two robots that are collaborating is challenging and even more so between a robot or human arm. The question of whether there is a way to model and plan the complex interactions between arms efficiently enough to be used to facilitate collaboration has yet to be answered. The answer to this question has applications in factory settings, personal robot assistance for motion and strength impaired, and more. Collaborative motion planning between two arms is hard because it is difficult to compute the exact location they need to be in order to complete a task. Previous research has successfully planned the motion and interaction of multiple robots and for robot manipulator arms solving complex problems with many degrees of freedom. Interaction templates is the solution to solving problems involving interactions between multiple robots [3]. Reachable volumes is the solution for motion planning for a manipulator arm with many degrees of freedom [4]. Although both of these solutions are good in respect to the problems they were created to solve, Interaction Templates on its own can't be used to solve problems involving a robot manipulator arm with many degrees of freedom, and Reachable Volumes on its own can't be used to solve problems involving many robots interacting with each other.

This project combines Interaction Templates and Reachable Volumes to model and to plan collaborative motions more efficiently between a human or a robot manipulator arm than either body of work could do on its own. Interaction templates are used to model possible interactions between a pair of robots. Each robot has an interaction template generated for them. The purpose of the interaction template algorithm is to use interactions in both necessary and optional cooperation problems by solving a motion planning problem. This is done by first creating a robot-type map for each robot-type, then connected all the different robot-type maps to each other with interaction edges in order to form a combined roadmap. A shortest path algorithm then can be used to find a path between a start and a goal in this combined roadmap.

The original way for creating Interaction Templates between two robots is to use a sampling method on each robot. Then each of the these created samples are added to the interaction template. This original way is flawed in that both arms don't have control over the other arm's end effector placement. This means that both arms are sampling independently from each other, and have to hope they both end up at the same location in order to do a handoff, which has a very low probability. The new way for creating Interaction Templates in this paper is to use one sampler for robot one. Using the end effector position of robot one as input, robot two is sampled so that its end effector meets robot one's end effector. This increases the probability from zero to closer to one hundred percent. After the Interaction Templates are created, they are used the same way as before to build the roadmaps and find a path between a start and a goal.

The new way for creating Interaction Templates is demonstrated in an experiment that involves two manipulator arms with two degrees of freedom. Both these arms performed a handoff in a virtual obstacle free environment in order to test the efficency and accuracy for the algorithms without obstacles. The new way had an order of magnitude decreased number of samples needed to perform a successful grasp. This resulted in the new way having a 90% success rate while the old way had a 10% success rate. These results show that the method that combines Interaction Templates and Reachable Volumes improve accuracy over prior work.

2. RELATED WORK

Collaborative robotic motion planning between multiple arms has already been researched in four different areas. These four different areas are Collaborative Planning with Arms, Probabilistic Road Maps, Interaction Templates for Collaborative Motion Planning, and Reachable Volumes for Constrained Motion Planning. The following sections will explain any previous research that been performed in these areas.

2.1 Collaborative Planning With Arms

Currently there has been some research on collaborative motion planning between robot arms. Researchers in Germany were able to do Task and Motion Planning by using a recurrent neural network that is then able to predict action sequences for motion planning. This neural network is able to do so by learning from an initial scene image [5]. The demonstration of this work was shown by having two robotic arms next each other, and they both were working together on the same goal which was to place a yellow box on a red target that is already occupied by another target. One arm could reach the yellow box, and the other arm could only reach the object that is on the red target. The researchers ran this experiment with multiple other objects in this scenario. The final conclusion for this experiment was that the neural network was able to learn from an initial scene image and apply it to a situations involving multiple objects, and that the neural network was able to execute with a high performance.

Other research that has been conducted for collaborative planning is combining high-level representation and causality-based reasoning with low-level geometric reasoning and motion planning [6]. This framework allows a program to calculate the shortest plans for complex planning and prediction problems with temporal constraints. The framework works by first finding an optimal task-plan. If there is no feasible kinematic solution for the task-plan found, then causal reasoner is guided by the motion planner. The motion planner directs the causal reasoner through modifying the planning problem with new temporal constraints. The causal reasoner also utilizes low-level geometric reasoning and kinematic relations in order to check for collisions. The framework in this research was applied by two pantograph robots working concurrently on a complex assembly task. This resulted in improved computational efficiency by a factor of 100. However, another conclusion from this research was that an accurate change in trajectory can't always be calculated in response to an uncertainty such as interventions and collisions with unknown objects.

Another piece of research on collaborative planning with robots was done on time-extended multi-robot coordination for domains with intra-path constraints [7]. Intra-path constraints are defined as constraints that impact route planning and occur on the paths of agents. This research applied two different methods for creating a coordinated solution under time constraints. One approach used tiered auctions and two heuristic techniques, clustering and opportunistic path planning while the other approach used a centralized genetic algorithm. Both these approaches were applied to the same scenario. The scenario consisted of fire trucks responding to multiple fire outbreaks across a city, however, there are impassable debris on the roads throughout the city. There are also bulldozer bots to push the debris out of the way. The genetic algorithm proved to be better for scenarios where performance is crucial, and time for planning is abundant. When there is less time available, the tiered solutions proves to be much better. The group plans to extend their work to situations with more uncertainty.

2.2 Probabilistic Road Maps

Research on motion planning began in the 1970s [8]. Original motion planning was shown to take exponential time to the number of degrees of freedom (DOFs) [9]. Degrees of freedom are the minimum number of parameters that are required to specify the state of a robot. An example would be a point in a coordinate plane has two degrees of freedom. During the 1980s, the concept of a configuration space was formed [10]. A configuration is defined as a complete specification of the position of every point in a robotic system. A configuration space is then defined as the set of all robot placements. Heuristic planners could be used to find a path in two and three dimensional C-spaces, but not any higher. In 1996, probabilistic roadmaps or PRMS were a breakthrough that enabled quick and reliable solutions to problems with many DOFs [11]. PRMs led to research in finding the shortest path in a configuration space instead of just seeing if a path exists.

Algorithm 1 describes the PRM steps. The algorithm starts by building a graph that models a space. The algorithm then randomly samples nodes that cite a specific criteria. The algorithm is then going to find a list of connections that could be added. Each connection is checked before being added to the roadmap. A shortest path graph algorithm is then used to search for a series of nodes and edges that traverse the roadmap to connect the start to the goal.

Algorithm 1: Algorithm to Create a Probabilistic Road Map (A, Q, W, P)

Ensure: Set of robot-types A, query pairs Q, interaction cost W, Planner P

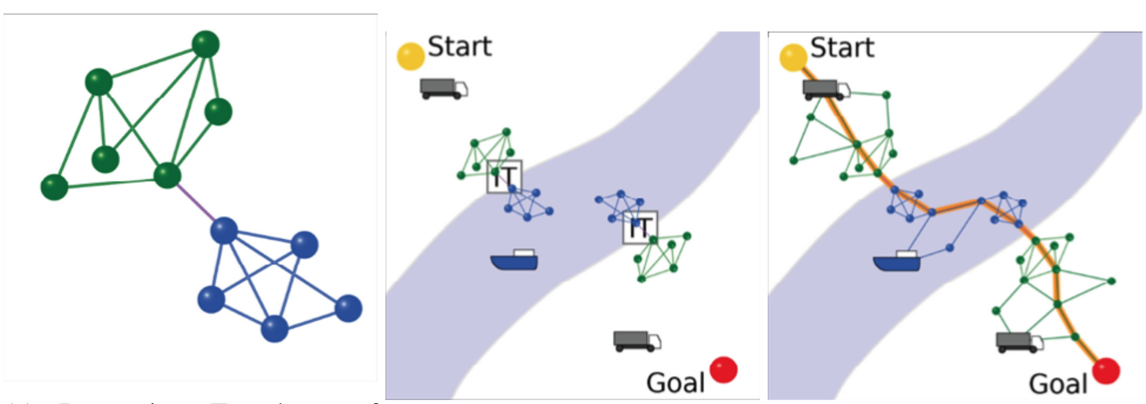
$R = (N, E) \leftarrow (\varnothing, \varnothing)$
while notDone() **do**
    $c \leftarrow RandomFreeCfg()$
    $N_c \leftarrow Neighbors(N, c)$
    $N \leftarrow N \cup c$
    for all $n \in N_c$, in order of increasing $\delta(c, n)$ **do**
        if $localPlanner.isConnected(c, n)$ **then**
            $E \leftarrow E \cup (c, n)$
        end if
    end for
end while
return R

2.3 Interaction Templates for Collaborative Motion Planning

Interaction Templates is a multi-robot planning method which moves certain robot types from the task planner to the motion planner [3, 12]. Interaction Templates have shown to have improved performance over current task and motion planning approaches. This better performance is due to Interaction Templates allowing road maps to only be calculated once for each robot in the system. Interaction Templates can be used between different types of robots for example a robot truck dropping off a payload to a robot boat.

An interaction template is a way to model possible interactions between any pair of robots in an environment. Inside each interaction template that are pairs of small roadmaps that are built around a pair of configurations that define the relative position of an interaction. An interaction edge is defined as a feasible robot state change in which there is possible task flow. An example of a robot state change would be the passing an object from one robot to the next.

The first step in the interaction template algorithm is to create a interaction template for each robot in the system which is shown in Figure 2.1a. This can be done by using PRM. The next step in the algorithm is to find possbile interactions between robots in both disjoint and overlapping workspaces. A disjoint workspaces would be defined as workspaces in which not all robots can move in. An example of this would be a car and a boat each having their own disjoint workspaces since each can't travel both on land and water which is shown in Figure 2.1b. An overlapping workspace would be just the opposite. Finding possible interactions in overlapping workspace make it easier to find a roadmap in a constantly changing topology. The next step in the interaction template algorithm is to transform the Interaction Template roadmaps for each given robot-type such that all the robots of the same type are part of the same roadmap. This can also be done using PRM. The final step is to create a combined roadmap that represents all the robots in the system. One could go from start to goal in the combined roadmap to see if an interaction can take place from a specific start and goal in the multi robot system which is shown in Figure 2.1c.



(a) Interaction Templates of Two Different Robots Connecting at the Purple Edge

(b) Possible Interactions Between Disjoint Workspaces

(c) Going from Start to Goal in a Combined Roadmap

Figure 2.1: Interaction Template Algorithm take from [13].

2.4 Reachable Volumes for Constrained Motion Planning

Reachable Volumes is a conceptual tool for quantifying the areas of a workspace a manipulator can access by creating a geometric representation of the regions the joints and regions can reach [14, 4, 15, 16, 2]. Figure 2.2 shows the Reachable Volumes for the joints of a fixed base manipulator. This is done by creating a new planning space called the RV-space which contains all points that fulfill a problem's constraints. This area of research is more pertinent to a robot with constraints are placed on a robot such as a robot moving and placing a binder into a binder holder.

The algorithm for reachable volumes places a joint every iteration until every joint is placed for a robot manipulator arm. Prior to joint placement, every joint has its full reachable volume that can be reached for that joint. After each joint is placed, a new constraint is made on the remaining joints' areas that haven't been placed yet. So the last joint to be placed should have a drastically smaller area that it can reach. Every time a joint is placed, its reachable volume for that joint is calculated using joints that have already been placed.

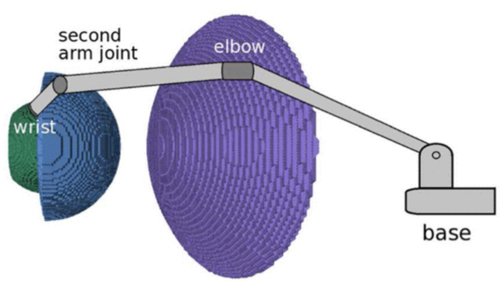


Figure 2.2: Reachable volumes for multiple joints and fixed base taken from [2].

3. METHODS

The prior way of multiple robot manipulator arms interacting with each other uses just Interaction Templates standalone, and each robot has no knowledge of the location of other robots. The probability of success for a handoff using this method is close to zero. The new way that is proposed in this research combines Interaction Templates with Reachable Volumes, and leverages knowledge of where one robot is placed to more easily find placements for other robots that can grasp at the same location. The prior way of multiple robots interacting will first be reviewed. Then the new way will be described and shown how it differs from the prior way.

3.1 Independent Interaction Template Algorithm

The prior way for creating Interaction Templates is to use a sampling method independently on each robot manipulator arm which is shown in Algorithm 2. The algorithm starts with creating a blank Interaction Template T, which is a small roadmap that captures how multiple robots can can collaborate in proximity to each other. Each robot is given a start, a, and goal, q, pair that describes the endpoints of the interaction. Each of these start and goal pairs are solved using a motion plan which creates a small roadmap that connects the start and the goal. This motion plan can be PRM. Then each of these smaller roadmaps are connected using their goals, f, in order to create a single roadmap which is the Interaction Template.

The problem with sampling independently is that each robot manipulator arm has no information about the location of the other arm. This results in each robot manipulator arm in the system to essentially guess where to meet other arms for an interaction. This results in a probability that is close to zero.

3.2 Dependent Interaction Template Algorithm

The solution in this paper is a Dependent Interaction Template Algorithm which is shown in Algorithm 3. It is called dependent because when the next robot is sampled, it depends on the output for the previous robots. The output can be any type of positioning of a robot. In the case

Algorithm 2: CreateInteractionTemplatesIndependent(A, Q, W, P)

Ensure: Set of robot-types A, query pairs Q, interaction cost W, graph-based motion planner P

1: create blank interaction template T
2: **for all** robot-type, query pair $(a_i, q_i) \in Q$ **do**
3:     **for all** robot-type, query pair $(a_j, q_j) \in Q$ **do**
4:         **if** i=j **then**
5:             **continue**
6:         **end if**
7:         Configure a_j at q_j.goal
8:     **end for**
9:     $r_i, f_i \leftarrow$ Solve motion plan for q_i for a_i with P
10:     $T.append(r_i, f_i)$
11: **end for**
12: **for all** $f_i \in T$ **do**
13:     **for all** $f_j \in T, j > i$ **do**
14:         Add edge in T with weight W, between f_i and f_j
15:     **end for**
16: **end for**
17: **return** T

of robot manipulator arms, the end effector position of previous robots is communicated to all following robots.

The algorithm that is created for the Dependent Interaction Template Algorithm differs from the Independent Interaction Template Algorithm. One way they differ is that there are multiple Planners which are represented by P_i where P_i is the planner for robot i. This allows for the dependent sampling. Another difference between the two algorithms is on line nine. The arbitrary start and its interaction goal is solved using the previous start and goal from another robot, allowing the dependent sampling to happen. In our solution for robot manipulator arms, P_0 is Uniform Random Free, and Reachable Volumes is P_1. This creates an Interaction Template algorithm that is tailored a lot more so for robot manipulator arms than the Independent Interaction Template algorithm.

The advantage of sampling dependently is that each robot manipulator arm has information about the location of the other arm. Sampling independently is equivalent to two people closing their eyes, and trying to guess where to put their arms in order to complete a handoff. The depen-

Algorithm 3: CreateInteractionTemplatesDependent(A, Q, W, P)

Ensure: Set of robot-types A, query pairs Q, interaction cost W, set of graph-based motion planners P

```
create blank interaction template T
for all robot-type, query pair (a_i, q_i) ∈ Q do
    for all robot-type, query pair (a_j, q_j) ∈ Q do
        if i == j then
            continue
        end if
        Configure a_j at q_j.goal
    end for
    r_i, f_i ← Solve motion plan for q_i for a_i with Planner P_i(r_{i-1}, f_{i-1})
    T.append(r_i, f_i)
end for
for all f_i ∈ T do
    for all f_j ∈ T, j > i do
        Add edge in T with weight W, between f_i and f_j
    end for
end for
return T
```

dent sampling solution is equivalent to one person moving their arm to one location first, then the next person moving their arm to the location of that person's arm. This increases the probability of success significantly and makes is closer to one hundred percent which is a notable difference from sampling independently.

4. RESULTS

In this section, we show the experiments that were conducted in order to compare two different methods, Interaction Templates standalone, and Interaction Templates combined with Reachable Volumes. The method that combines Interaction Templates with Reachable Volumes also uses the location of one robot in order to calculate the location for a handoff with other robots. All experiments were in a virtual environment that could simulate two robot manipulator arms performing a task. Each robot was tested on using accuracy of doing a handoff between the two different arms by measuring the number of samples that each sampler takes in order to get both manipulators in the position of a handoff. The expectation of the results is that the combination of Interaction Templates with Reachable Volumes will perform significantly better than Interaction Templates each on its own.

4.1 Experimental Setup

A virtual environment was setup using an application called Vizmo that was created by the Parasol lab. This virtual environment is essentially a 3-dimensional box that contained nothing but the two arms. Inside this environment, there are two robot manipulator arms that were used to do a handoff. Each robot manipulator has a fixed base and two joints, so each arm has two degrees of freedom as shown in Figure 4.1. There were no obstacles in this environment because the goal of the experiments was to test the efficiency and accuracy of the algorithms without obstacles.

One theme of the experimental setup was to keep the input parameters as similar as possible between the different methods. This is important because in order to accurately compare each method to each other, the same information should be passed to each method. The two robots used in this experiment, were kept in the same positions for both trials that involved independent and dependent sampling.

One method that is being experimented with is independent sampling in which both samplers for both robot manipulator arms are unconstrained. The sampler uses uniform random free

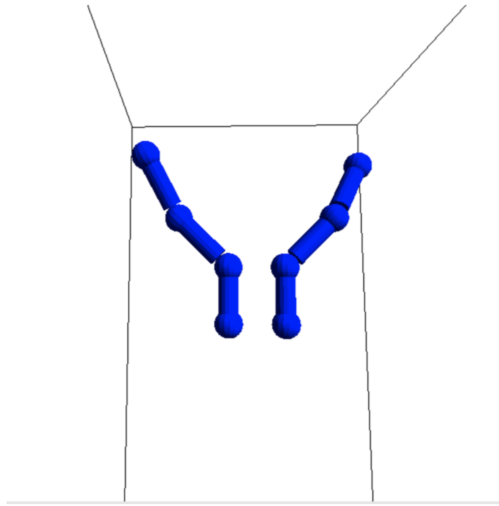


Figure 4.1: Two robot manipulator arms in a virtual environment.

sampling. Because the sampling is independent, either robot doesn't have any information on the location of the other robot manipulator arm. This method was included in the experiment since it is the prior way of sampling Interaction Templates.

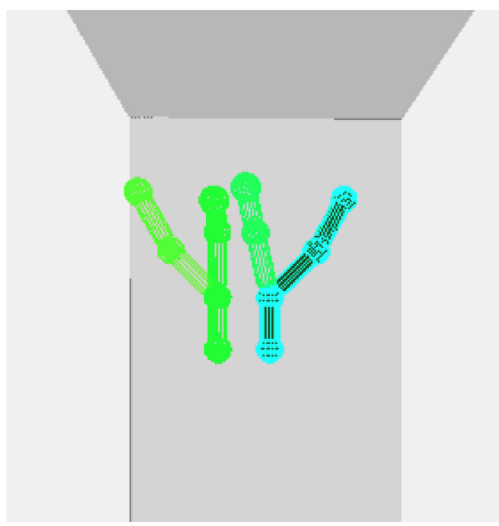


Figure 4.2: Two robot manipulator arms joining together to do a handoff.

The second method that is being experimented with is the new or dependent way of sampling interaction templates for robot manipulator arms. This method has one robot manipulator

Table 4.1: Independent and Dependent Sampling Compared

Sampling Method	Independent Sampling	Dependent Sampling
Number of Nodes in Map	40	42
Number of Edges in Map	370	344
Number of Attempts	931.5	1
Trials that Successfully did a Handoff	10%	90%
Min Distance Between End Effectors	1.49	0.32
Max Distance Between End Effectors	22.8	0.32
Avg Distance Between End Effectors	12.59	0.32

arm that is unconstrained. The meaning of unconstrained in this situation is that the first sampler is free to move anywhere. The second manipulator arm is constrained by Reachable Volumes. Reachable Volumes closes the area in which one arm can reach by forming a small box. Also the second manipulator arm would also be receiving information on the location of the first manipulator arm since this method is dependent.

4.2 Experimental Results and Discussion

Seven different metrics were used to compare the two different sampling methods on the two arms doing a handoff as in Figure 4.2. One metric was the size of the map, which is made up of nodes and edges, that is created after the Interaction Template algorithm. This metric was measured in order to see which type of sampling uses less memory. The second metric that was compared between the two different sampling methods was accuracy in order to see which method is the most successful. Accuracy is to be shown through the number of attempts for the two arms in a trial run to find a handoff, and the percentage of trials that had a successfuly handoff. The last three metrics recorded were the minimum, maximum, and average distance between the end effectors for all samples taken. These were recorded since they show the precision of the samples taken.

Referring to the results of the experiments in Table 4.1, the number of nodes and edges of the independent and dependent sampling methods are marginally different. A more noticeable difference is seen in the number of attempts of independent sampling, which is 931.5, and the number of attempts for dependent sampling, which is 1. This comparison shows how much more accurate dependent sampling is due to the passing of one end effector location to the other. Also the success of a handoff for dependent sampling is much higher at 90 percent when compared to the success of a handoff for independent sampling which is at 10 percent. This further shows the increase in accuracy that comes with dependent sampling. Also the minimum distance between the end effectors for independent sampling was much higher at 1.49 when compared to dependent sampling at 0.32. The significant distance between the distances can also be seen between the max and average distance of independent sampling, 22.8 and 12.59, and the max and average distance of dependent sampling, 12.59 and 0.32. dependent sampling having a much lower distance seen between end effectors can be attributed to the Reachable Volumes sampling a smaller space when compared to the Uniform Random sampling that was used for independent sampling.

5. CONCLUSION

A method to solve interactions between robot arms, or between a human and a robot arm in constrained spaces such as those in a factory is presented by combining the solution to constrained spaces, Reachable Volumes[2] and the solution to multiple interactions, Interaction Templates[3]. This integration is done by replacing the motion planning algorithm that is utilized in Interaction Templates with the algorithm for Reachable Volumes. This new algorithm is referred to as dependent sampling because the end effector position of one robot is passed to the other robot which creates a more accurate algorithm.

An experiment was conducted to show the differences between the dependent and independent sampling between two different manipulator arms with two degrees of freedom. The dependence that is instrumental to dependent sampling proved to have a significant increase in accuracy in performing a handoff by decreasing the number of attempts to find the location to do a handoff, and increasing the success of two arms finding a handoff. The Reachable Volumes also showed to decrease the area that the manipulators sample in order to do a handoff which is shown in the min, max, and average distances between end effectors. In conclusion, the dependent sampling presented in this paper is more accurate than independent sampling with Interaction Templates.

6. FUTURE WORK

More experiments can be done in the future to show that the algorithm for Reachable Volumes and Interaction Templates improves collaboration among multiple robot manipulator arms. An experiment that involves three or more robot manipulator arms can be implemented to show improvement in speed and memory for handoffs with more than two arms. This method could certainly be extended to be used between multiple arms by having a chain reaction among manipulators by having each arm send its location of its end effector to the next arm in the handoff. Another experiment that can be apply to the method in this research is to have both robot manipulator arms to have fixed bases. Both manipulators having fixed bases would be similar to the manipulators that are used in factories to piece together a car on an assembly line. Reachable Volumes has to be improved upon to work with manipulators with fixed bases.

Another way that this research method can be improved is able to sample around different constraints. Currently the Reachable Volumes and Interaction Templates method are only used for sampling with arms that are of a fixed size. There are real world applications for the manipulators to be able to work with humans such as personal robot assistance for motion and strength impaired. Different humans can have arms of various lengths, which provides a harder challenge for the method that is presented. Another challenge rises when more degrees of freedom are added to arms. This creates a larger area that is needed to sample, thus making it more complex to find a handoff between arms.

REFERENCES

[1] J. C. Latombe, "Motion planning: A journey of robots, molecules, digital actors, and other artifacts," *Int. J. Robot. Res.*, vol. 18, no. 11, pp. 1119–1128, 1999.

[2] T. McMahon, S. Thomas, and N. M. Amato, "Sampling based motion planning with reachable volumes for high degree of freedom manipulators," in *Int. J. Robot. Res.*, vol. 37, pp. 779–817, June 2018.

[3] J. Motes, R. Sandström, W. Adams, T. Ogunyale, S. Thomas, and N. M. Amato, "Interaction templates for multi-robot systems," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2926–2933, 2019.

[4] T. McMahon, S. Thomas, and N. M. Amato, "Sampling based motion planning with reachable volumes: Application to manipulators and closed chain systems," in *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, (Chicago, Il.), pp. 3705–3712, Sept. 2014.

[5] M. T. Danny Driess, Jung-Su Ha, "Deep visual reasoning: Learning to predict action sequences for task and motion planning from an initial scene image," *Robotics Science and Systems*, p. 3, 2020.

[6] E. Erdem, K. Haspalamutgil, C. Palaz, V. Patoglu, and T. Uras, "Combining high-level causal reasoning with low-level geometric reasoning and motion planning for robotic manipulation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 4575–4581, IEEE, 2011.

[7] E. G. Jones, M. B. Dias, and A. Stentz, "Time-extended multi-robot coordination for domains with intra-path constraints," *Autonomous robots*, vol. 30, no. 1, pp. 41–56, 2011.

[8] N. Nilsson, "A mobile automaton: An application of artificial intelligence techqniques,," in *1st International Joint Conference on Artificial Intelligence*, 1969.

[9] J. H. Reif, "Complexity of the mover's problem and generalizations," in *Proc. IEEE Symp. Found. Comp. Sci. (FOCS)*, (San Juan, Puerto Rico), pp. 421–427, Oct. 1979.

[10] T. Lozano-Perez, "Spatial planning: A configuration space approach," *IEEE Trans. Comput.*, vol. 100, no. 2, pp. 108–120, 1983.

[11] L. Kavraki, M. Kolountzakis, and J.-C. Latombe, “Analysis of probabilistic roadmaps for path planning,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 4, pp. 3020–3025, 1996.

[12] J. Motes, R. Sandström, H. Lee, S. Thomas, and N. M. Amato, “Multi-robot task and motion planning with subtask dependencies,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3338–3345, 2020.

[13] J. Motes, R. Sandström, W. Adams, T. Ogunyale, S. Thomas, and N. M. Amato, “Interaction templates for multi-agent systems,” Tech. Rep. TR18-002, Parasol Lab, Dept. of Computer Science and Engineering, Texas A&M University, August 2018.

[14] T. McMahon, S. Thomas, and N. M. Amato, “Sampling based motion planning with reachable volumes: Theoretical foundations,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, (Hong Kong, China), pp. 6514–6521, May 2014.

[15] T. M. McMahon, *Sampling Based Motion Planning with Reachable Volumes*. Ph.D. dissertation, Dept. of Computer Science and Engineering, Texas A&M University, Aug. 2016.

[16] T. McMahon, R. Sandström, S. Thomas, and N. M. Amato, “Manipulation planning with directed reachable volumes,” in *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, (Vancouver, Canada), Sept. 2017.