COLLABORATIVE ROBOTIC MOTION PLANNING

An Undergraduate Research Scholars Thesis

by

THOMAS COUSINS

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 2022

Major: Computer Engineering

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, Thomas Cousins, 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

Thomas Cousins
Electrical and Computer Engineering
Texas A&M University

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

When attempting to plan for an interaction between two robots there are many factors that compound its complexity, including the number of joints each robot has, causes for collision, power usage, placement of the end effectors, and others. These factors compound themselves as the robots involved increase in joints, there is an increase in obstacles, or more robots are added to the system. In an effort to devise an algorithm that provides a fast and effective solution to these types of problems, previous work has been done on Reachable Volumes and Interaction Templates, algorithms that serve to reduce the processing time for high degrees-of-freedom (dof) robots with stringent constraints, and to reduce the processing time for certain interactions between two or more systems of robots, respectively. Reachable Volumes uses the volumes that each joint of the robot can occupy, and by choosing points within these volumes it can solve for a configuration at a given end effector position. This enables a quick way to evaluate a configuration for a high dof robot as well as providing a means of avoiding collisions with obstacles by simply sampling the volumes around the joints in areas not occupied by the obstacles. Additionally, another strength of Reachable Volumes is it provides the ability to precisely place the end effector at a desired location, which other motion planning algorithms struggle with. Interaction Templates can be used to "template" or formulate an interaction such as a handoff that can be applied to a "roadmap" or a

collection of points that make up the movements of a manipulator in configuration space. In doing so, Interaction Templates can make complex and processing expensive tasks such as calculating for a handoff interaction much more efficient. The work below begins to integrate elements from the Reachable Volumes algorithm into Interaction Templates, which speeds up several steps on the Interaction Templates Algorithm as well as makes it more effective for high dof robots or situations with a multitude of constraints. It additionally sets up a framework for more robots to be included in the algorithm.

ACKNOWLEDGMENTS

Contributors

I would like to thank my faculty advisor, Dr. Shawna Thomas, and 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.

The software analyzed/used for this project were provided by Dr. Thomas, James Motes, and Parasol Labs.

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

Collaborative robotics is essential towards boosting efficiency and productivity in an economic environment that is becoming increasingly automated. Multi-robot systems play a significant role in the automatization of industries and can provide noticable advantages over single-robot systems. This includes handoff-based tasks, where a series of robots perform an operation and then hands off an object, to tasks where there are multiple robots working in the same area at the same time. For the former, a good example would be the transition of partially assembled car parts from one workspace to another. For the latter, an example would be having a grouping of robots working together to lift an object. However coordinating multi-robot systems is a complicated and detailed task because each robot is not necessarily aware of the other, and their interactions can change as different tasks are performed. For instance, in the car parts example mentioned earlier, collisions during the handoff process can occur, and, dependent on the part being transferred, the handoff process may need to change. Current implementations for interaction algorithms which can solve these types of operations are cost-heavy and require additional calculations to account for multi-robot collisions. These algorithms also do not scale well with higher degrees of freedom robots with varying joint types.

Inter-robot tasks are difficult to coordinate for a number of reasons. There needs to be a low degree of error to ensure efficient and accurate movements. In addition, the robots need some degree of awareness of each other to perform this interaction. If the degree of error is too large, the robot may be in a position such that the task is not being properly performed, or there is a collision chance between the robots. This is particularly difficult the more joints that they might have, as that is another dimension that needs to be accounted for during the motion planning stage. Because of this, typical motion planning algorithms take an exorbitant amount of time in order to plan these interactions.

As a method for increasing efficiency and scalability of collaborative robotics calculations, previous work on Interaction Templates (IT) [1] has shown a marked improvement on the calculation time required for the interactions. Interaction Templating provides a small roadmap that guides these interactions, which are generated with standard motion planning algorithms beforehand [2]. These ITs are then connected to form a combined roadmap to fulfill the task. IT's are a powerful tool because they account for collisions during the motion planning stage and are far more scalable than conventional algorithms. However, IT's are not efficient at computing the viable configurations of an end effector such that it is close enough to another end effector so that an interaction can occur. Additionally, the motion planner within IT's is still dependant on the speed of traditional motion planning algorithms, increasing the preprocessing time.

Previous work has also been done on Reachable Volumes, which can be used to calculate configurations given constraints. The main appeal of Reachable Volumes is that not only is it quicker than standard configuration algorithms for instances with high degrees of freedom, but it can work with unique robot designs [3]. Additionally, Reachable Volumes is capable of precisely placing components such as the end effector, a necessary trait when attempting to plan an interaction. Reachable Volumes can also be used to sample with constraints such as obstacles easily by reducing or removing volumes from the volume around the joints. For instance, if an obstacle is in a known location, all regions of the volumes around the joints of the robot in intersection with the obstacle can be discounted before sampling. While Reachable Volumes is fast, it is unscalable for instances where the configuration has many solutions, as it would have to take into account interaction motions at every place the planner considers.

The intention of this research is to implement Reachable Volumes during the motion planning stage of the Interaction Templates algorithm in order to make it more effective for high degrees-of-freedom robots, quicker, and more precise. This will enable the Interaction Templates algorithm to complete faster, reducing the time spent in pre-processing for the motion planning as well as be capable of planning for interactions that might require more accurate positioning. This algorithm will not only be beneficial towards high degrees-of-freedom robots, but will have improvements across the board as well as be capable of working with large quantities of robotic entities. Additionally, by using the Interaction Templates framework, planning for a significant quantity of robots and their interactions becomes far more feasible.

The results of this research by the time of this thesis being written got as far as adding more than two robots into the same Interaction Template using the Independent Sampling technique. Further work is needed to modify the Dependent Sampling technique as well as make this applicable to manipulators with large degrees of freedom. The different sampling techniques are discussed in the Methods section.

2. RELATED WORK

This work has been based off of several concepts, including Sampling-Based Motion Planning and Motion Planning for Multiple Robot Collaboration. Sampling-Based Motion Planning is a technique for robotic motion planning that involves sampling a region to create nodes, which are then connected via edges to create a roadmap which can be used to traverse an environment [4]. Motion Planning for Multiple Robot Collaboration is an all-encompassing term that applies to any system with two or more robots that are operating within the same region [5].

Additionally, there are two algorithms that underlie the work done in this paper, Reachable Volumes and Interaction Templates. Reachable Volumes is a unique algorithm for calculating robot configurations that is strong for robots with many joints as well as problems with a large degree of constraints [6, 7, 8, 9, 10, 11, 12, 3, 13, 14]. Interaction Templates are an algorithm used to speed up processing time for common interactions between robots by "templating" them, which allows some processing steps to be skipped when attempting to coordinate an interaction [15, 16].

2.1 Sampling-Based Motion Planning

Sampling-based motion planning is one of the most common approaches towards robotic motion planning. In particular, the method used in this work is PRM or "Probabilistic Roadmaps." The basis for these models of motion planning is by sampling the configuration space to generate valid points, or nodes, that are collision free, then connecting those points via edges to generate a roadmap [17]. The configuration space can be defined as the set of all possible robot configurations, containing the same volume as the motion space but with a higher dimensionality equivalent to the degrees of freedom of the robot [18]. The motion space can contain obstacles, structures, and other robots, and is the 3D volume in which the robot will be operated. The degrees of freedom are equivalent to robot parameters such as joint angles or position, so the configuration space is effectively written as a combination of these joint angles that positions the robot and its components in a certain fashion. The points that are sampled for the roadmap are written in terms of the configuration space, meaning they are specific to one unique configuration of the robot. The edges that then connect these points contain the information required to go from one configuration to another.

There are various techniques for sampling the configuration space to generate a roadmap, including gaussian and uniform random sampling [19]. Gaussian and uniform random sampling are among the more common ones, and the ones that are used for the majority of this project. Gaussian sampling is distributed such that the nodes lie closer to obstacles, whereas random sampling samples the configuration space randomly, which tends to sample more evenly. When these samples are made, they are then checked to make sure they do not reside in collision with any object before they are added to the roadmap.

These points are then connected via "edges," which contain the information required to travel from one point to the next. The edges between points are selectively generated through algorithms such as the nearest neighbor, which will create an edge between a point and the closest point to it. This ensures that there is always a way to arrive at or near any point that has been sampled in the configuration space by traversing these edges. Of course, just a simple nearest neighbor algorithm can be limiting in terms of available routes and will not always arrive at an ideal path, so there are addition considerations and algorithms used when deciding where to generate these edges. Additionally, just like the nodes, the edges are checked to determine if they are in collision with any obstacles before they are added to the roadmap.

The roadmap that is generated can then be used to quickly generate a route to a point by the motion planning algorithm. In order to do so, the motion planning algorithm generates a configuration of the goal location and connects it to the nearest node. This is done for the start configuration as well. The roadmap is then traversed in order to find the lowest cost route from the starting node to the ending node. The edges along that route then become the path that the robot will follow for the task. This drastically saves on calculation time, as the edges are already pre-calculated, so those movements no longer become something that needs to be accounted for.

2.2 Motion Planning for Multiple Robot Collaboration

When multiple robots are added to an environment, the complexity of planning a task vastly increases. The problem now contains not only an additional configuration space, but additional factors that need to be taken into account including collision between two robots as well as sampling such that interactions between the robots can be facilitated. These systems are typically handled in one of two ways, with a centralized coordinator distributing tasks [20], or using a decentralized method which relies on distributed algorithms to navigate the robots [21].

There are a large quantity of currently existing algorithms that serve to solve for these tasks including using neural networks on individual robots and evolving them before combining them to solve a task [22], using social potential fields [23], and using a scheduling algorithm to ensure no collision [24]. However, each of these techniques have their own failings which prompted this research. Neural networks are capable of having complex and thorough mappings of a region, however they require a "learning" phase that can take a long time as well as are memory intensive. Social Potential Fields are similar to force vectors and are typically applied to robots systems that are very high in number. As such, social potential fields are too generalized for more complex tasks such as handoffs or precision movements. Scheduling algorithms are the theoretically best solution, however they typically require processing power during the tasks as well as can cause collision and deadlock. Additionally, as the number of joints or manipulators scales upwards these algorithms find it drastically harder to compute a solution.

2.3 Reachable Volumes for Manipulators

Reachable Volumes is a method used to generate points within a configuration space using the volumes around each joint within a robot [6, 7, 8, 9, 10, 11, 12, 3, 13, 14]. It does this by computing the reachable volume of the robot using the Minkowski sum. It then places the end effector at the desired location and then works backwards to generate positions for the remaining joints by sampling the volumes around each joint. This is a unique form of sampling that supports more diverse robot shapes, such as linkages, high degree of freedom robots, and tree-like robots

[14].

This technique comes with many strengths including speed, the ability to work with high degree of freedom robots, and the ability to operate under varied and heavy constraints without a speed concern [25]. The types of constraints Reachable Volumes supports includes end effector constraints, internal joint constraints, and multiple joint constraints [14]. Reachable Volumes is also capable of working with a large variety of joint types, including planar, spherical, prismatic, and combinations [14]. Reachable Volumes adaptability, precision, and ability to operate under heavy constraints are what make its combination with Interaction Templates particularly valuable. This is because, in order to facilitate interactions, the points generated during the motion planning stage need to follow precise positioning such that the end effectors can be within range of each other and have the other components of the robot out of collision. As mentioned above, Reachable Volumes is also very fast, and was shown to run in O(J*diameter(R)) time, where diameter(R) is the diameter of the robot and J is the number of joints. Figure 2.1 below illustrates the joint volumes used for the Reachable Volumes algorithm for a manipulator.

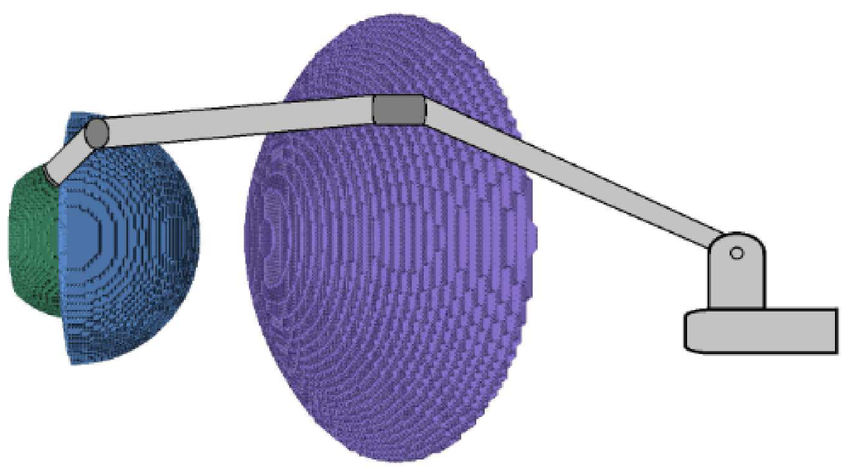


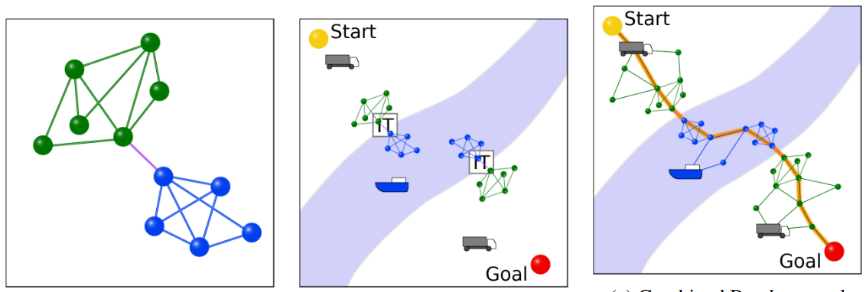
Figure 2.1: The reachable volume of three constrained joints grasping an object [12]

2.4 Interaction Templates

Interaction Templates are a technique that can be used to govern all of the inter-robot interactions that will occur during the planning stage [15, 16]. Interaction Templates offer speed-ups over traditional multi-robot motion planning by "templating" certain interactions, such as hand-offs, and then applying them where necessary as opposed to performing expensive calculations for every interaction. A "template" is essentially a pre-computed interaction which generalizes the processing-heavy task of calculating an interaction.

These templates can be duplicated and connected where applicable to build a more comprehensive roadmap, connecting roadmaps of one or more robots. These paths are only transversable using the paths of the robots involved in the interaction. In order to choose which robots form these interactions, it "auctions" off the individual subtasks that compose an interaction. During the auctioning process, robots are tested for corresponding capabilities to the task and then assigned to the task if they are capable of performing it. For some tasks that involve multiple robots, each role is auctioned off.

Assume that there is a scenario in which it is desired to transfer a large object across a river to a location on the other side. Ordinarily, had there not been a river in place, this could simply be done by loading the object into the back of a truck and driving to the desired location. However, now that the river is in place there needs to be a structure that allows for the transfer of the object from the truck to something that can cross the river, such as a boat or other vessel. Next, there needs to be another exchange from the boat to another truck which would then transport the object to its final location. This scenario and how Interaction Templates solve it is demonstrated in Figure 2.2. Firstly, an Interaction Template is formed governing the handoff from a truck to a boat and vice versa. This is then placed in the requisite positions for the handoff from truck to boat and the handoff from boat to truck using "tiling". By using the Interaction Templates in this manner a combined roadmap consisting of both trucks and the boats roadmap is formed. Now, a path can be derived from which the box can arrive at any location serviceable by the trucks, boats, or any combination thereof. By using Interaction Templates in this example, time was saved by avoiding the costly processing steps that would be required to calculate the truck to boat and boat to truck interactions.



(a) Interaction Template

(b) IT Tiling

(c) Combined Roadmap and Extracted Task Plan

Figure 2.2: Interaction Template Example Image from [15]

Interaction Templates vastly saves in time by short-cutting many intensive calculations and has shown a drastic reduction in time during the motion planning stage when compared to FFRob, a common task and motion-planning technique for multi-robot interactions [15]. FFRob is an algorithm that extends upon the commonly used "Fast-Forward" algorithm by taking into account reachability during the motion planning process [26].

3. METHODS

3.1 Materials

The materials involved in this research were a testing environment provided by Texas A&M University, Vizmo, which is a viewing environment capable of viewing the calculated paths of the robots, and starting code provided by Parasol Labs. Parasol Labs has conducted much prior research and work on sampling algorithms including Reachable Volumes and Interaction Templates.

3.2 Procedure

Reachable Volumes code was provided from Parasol Labs using a Git service, as was the base code for Interaction Templates. In order to combine them, Reachable Volumes was added into the sampling stage of the Interaction Templates code. It was done such that the first manipulator would sample the area for the handoff to occur, and then using those points and a volume around them corresponding to the handoff distance, Reachable Volumes would calculate valid configurations for the other manipulators involved in the task. This was also known as Dependent Sampling, as robot configurations were based upon each other. The pseudocode below in Algorithm 1 outlines this procedure. In the big picture, this can be used in order to facilitate handoffs and other templated interactions between higher degrees of freedom robots more effectively, while also allowing for flexible handoff positions by sampling relative to a leader that creates the seed positions.

Additionally, there is a method of sampling that does not sample robot's positions relative to each other, but rather samples them independently of each other, known as Independent Sampling. The pseudocode for this technique is shown below in Algorithm 2. This technique can be prone to collisions and requires a large amount of sampling to form non-colliding samples. This was the first to be implemented, and serves as the proof that many robots can be used for an Interaction Template. The procedure outlined by the Independent Sampling method was then used as a baseline to develop the Dependent Sampling method.

In order to test on a series of robots, or systems with multiple robots, a way to add several manipulators into the currently existing framework was needed. This involved using robot "groups" that contained objects that were linked together by some relationship. It also required the creation of "stages" which state the groups existing at a given moment in time. As the interaction progresses, the stages change, and these groups change depending on what objects are in contact or related to each other.

For instance, consider a situation in which a robot carrying a box transfers that box to two other robots which would then carry the box together. This would begin with a group "A" which would consist of a robot "1" holding a box, and another group "B", consisting of two robots, "2" and "3". These groups would then form the initial stage, "Start Exchange". The next stage would consist of the robots "2" and "3" carrying the box object, which creates another group "C". This then leaves the robot that initially held the object, "1", in a group by itself, "D". These two groups, with robots "2" and "3" carrying the box, and robot "1" alone, would make up the final stage "End Exchange". The system would then go from the first stage, "Start Exchange", to the next stage "End Exchange", which contains all the necessary information governing the transference of the box from group "A" to group "C".

The two methods of sampling mentioned earlier, Dependent and Independent, would be used in the interaction above to actually provide valid configurations for both the robots within a group and for the groups relative to each other. This means providing valid configurations for the robot "1" and box such that the box being held by robot "1" as well as making sure those configurations are close enough to the group "B" to facilitate the handoff in the next stage.

Using the modified code based on the pseudocode below and a handoff operation involving three or more robots, testing could then begin. Testing was done on simple proof of concepts for the Independent Sampling model.

3.3 Dependent Paths

The pseudocode below in Algorithm 1 illustrates the "Dependent Paths", or "Dependent Sampling" method for implementing Reachable Volumes in the Interaction Templates algorithm. What differentiates this algorithm from the "Independent Paths" or "Independent Sampling" method is that the sampling is done relative to a "leader". This is shown in the SeedPoints array, as every point sampled for the first robot, which is assumed to be the leader, is appended to this array and the following robots need to sample within a volume around each of the leader's points. The volumes around each point are generated around the time of sampling and are stored in the SeedBounds array. By passing these bounds into the sampling for the other robots, it is guaranteed that the points will be within the boundary specified. Typically, this boundary is set to an allowable radius for the interaction to occur. By using a multitude of SeedPoints, there is a variety of locations from which a handoff can occur. Once these samples are generated, the nodes are connected and edges are drawn up between them.

There are several purposes to this sampling method including increasing the connectivity of the roadmaps by providing a large amount of locations where an interaction can occur, which also diversifies the solution set, as well as being more efficient and effective as the pool of valid samples becomes smaller. This allows for a more intricate combined roadmap that supports several different ways of performing the same interaction. Additionally, when done this way, there is potentially less time spent in the sampling phase, as while Independent Sampling is akin to taking a random guess and hoping that the selected points will be valid, Dependent Sampling already takes these constraints into effect at the time of sampling. For interactions with few robots and sampling locations this difference is not as drastic, however as the robots involved in the operation increase, or the constraints become more stringent, Independent Paths has a hard time choosing a valid random location whereas Dependent Paths fares far better.

Algorithm 1 PseudoCode for implementing RV in IT using Dependent Paths

```
procedure GENERATEDEPENDENTIT(Tasks T, Robots R, Sampler S, Connector C, HandoffDistance H, NumNodes NN. NumAttempts NA)
    for Every Task t in T do
        FirstTime=True;
        SeedPoints = [];
        SeedBounds = [];
        for Every Robot r in R do
            if r.isCapable(t) then
                if FirstTime==True then
                    FirstTime=False;
                    SeedPts = S.sample(r, numNodes=NN, numAttempts=NA);
                    for Point in SeedPts do
                        SeedBounds.append(BoundingSphere(Radius=H, Center=Point));
                    end for
                    IT.AddNodes(SeedPts);
                    Connections=C.Connect(SeedPts);
                    IT.AddEdges(Connections);
                else
                    for Index, GoalPoint in SeedPts do
                        Points=Sample(numNodes=NN, numAttempts=NA, Robot=r, boundary=SeedBounds[Index], sampler=ReachableVolumes);
                        IT.AddNodes(Points);
                        Connections = C.Connect(Points);
                        IT.AddEdges(Connections);
                    end for
                end if
            end if
        end for
        Return IT
```

3.4 Independent Paths

The pseudocode for sampling using Independent Paths is shown below in Algorithm 2. Note that this algorithm does not take into account a volume around the points and just samples randomly instead. If an invalid point is sampled it simply samples again. This technique is fine for simpler implementations with a large degree of configurations available, but as the operation becomes more complex, either by adding more robots or constraints, the chances of randomly selecting a point that is valid drastically goes down. A configuration is considered valid if it meets the requirements for an interaction, namely being close enough to each other, and is not in collision. The distance can be fixed by setting up a boundary in which all sampling must occur, which is labeled as "HandoffBounds" in the algorithm. However, when sampling, collisions might occur which would cause the "Sample" function to run longer until it can generate a valid sample.

Algorithm 2 Pseudocode for Independent Paths

```
procedure GenerateIndependentIT(Tasks T, Robots R, Sampler S, Connector C, HandoffBounds H, NumNodes NN. NumAttempts NA)
    for Every Task t in T do
        for Every Robot r in R do
            if r.isCapable(t) then
                Points=Sample(numNodes=NN, numAttempts=NA, Robot=r, boundary=HandoffBounds, sampler=ReachableVolumes);
                IT.AddNodes(Points);
                Connections = C.Connect(Points);
                IT.AddEdges(Connections);
            end if
        end for
    end for
    Return IT
```

4. RESULTS

4.1 Experimental Setup

This project was done building off of previous work for motion planning provided by the University of Illinois. Within this work, the algorithm for both Reachable Volumes and Interaction Templates also existed. Additionally, this programming environment consisted of several files describing the robots, environments, and problem. These files included environment files which contained the obstacles, if there were any, and the boundary of the environment as well as files which included descriptions of the robots such that they could be recognized by Vizmo. The .xml problem files contained descriptions which outlined many aspects of the problem attempting to be solved, including paths to the robots, a path to the environment, which samplers to use, the groups and stages, and other factors that needed to be accounted for during solving.

4.1.1 Materials

In order to test the algorithm as well as implement it, several resources are needed. This includes an environment which contains the details of the robots and obstacles in the space in which the algorithm is running. A problem file containing information on the problem was necessary as well. Additionally, a way to compile this environment into something containing a solution that is viewable in Vizmo for the robots and their movements is required. Code that contains the implementations of Interaction Templates and Reachable Volumes is also needed for this project. These items are all contained within a repository provided by the University of Illinois, however many of these were modified over the course of the project. Modifications included the creation of an empty environment file, the modification of the .xml file describing the problem, and the modification or addition of the dependent and independent sampling algorithms. Vizmo was used to view the roadmaps and path files once they were generated by the aforementioned compiler that was provided by Parasol Labs.

4.1.2 Procedure

First, an environment consisting of three robots and no obstacles was generated, for testing. The robots populating this environment are simple, however they are unique. A roadmap for this environment is then generated using the "independent" sampling method in order to generate the interaction templates. This roadmap will then serve to prove that the current methods can be scaled to include upwards of two robots. The next step involves editing the "dependent" method for generating Interaction Templates in order to not only make sure it is scalable to a large quantity of robots, but also to implement Reachable Volumes during the sampling stage. Reachable Volumes will be used in the dependent technique to generate valid configurations for interactions to take place, through using initial "seed" locations and then sampling for the other robots based upon this "seed." As of the time of writing this thesis, only independent sampling has been scaled up to three robots and is discussed below, and is done using a uniform random sampling method.

In an example test, Independent Paths was used to solve the box handoff process mentioned in the methods section. This example had a robot holding a box, then passing it to two other robots which would both carry the box. This had the problem broken up into several stages and groups beyond what was mentioned in the methods section. This problem additionally only had certain regions in the environment where each robot could operate, which can be likened to the boat and truck example listed in Section 2.4. The first stage was the Initial Stage, which was the starting positions for each robot. This then transitions to the EnterExchange stage, which still has Robot 1 from the example holding the box object. The next phase is the ExitExchange stage, which now has Robot 1 by itself and Robots 2 and 3 supporting the box. The last stage is Final and contains the positions of the robots after the handoff is completed. The roadmaps were generated for each stage for each robot. The robots are labeled as "Boxy1", "Boxy2", "Boxy3", and "Passive". Boxy 1-3 are the robots that are holding the Passive, or "box" as it is mentioned in the example, object. Some screenshots of the roadmaps from the EnterExchange stage are shown below. Several of these only contained one or two nodes due to the simplicity of the problem.

4.1.3 EnterExchange Roadmaps

These roadmap screenshots all come from the EnterExchange stage of the handoff operation. During this stage, Robot1 is represented by Boxy1, which is a simple box-shaped robot. The same applies to robots 2 and 3, which are also box shaped robots with no joints, labeled Boxy2 and Boxy3, respectively. The passive robot represents the object being handed over, and it is also a simple shape similar to the boxy robot's shapes. During the EnterExchange phase, we see that the Passive object has a roadmap containing three points. Figure 4.1 illustrates the various positions that it can or will move through during the "EnterExchange" stage. It is useful to note that at this point in time, the passive object is still tied to the Boxy1 robot.

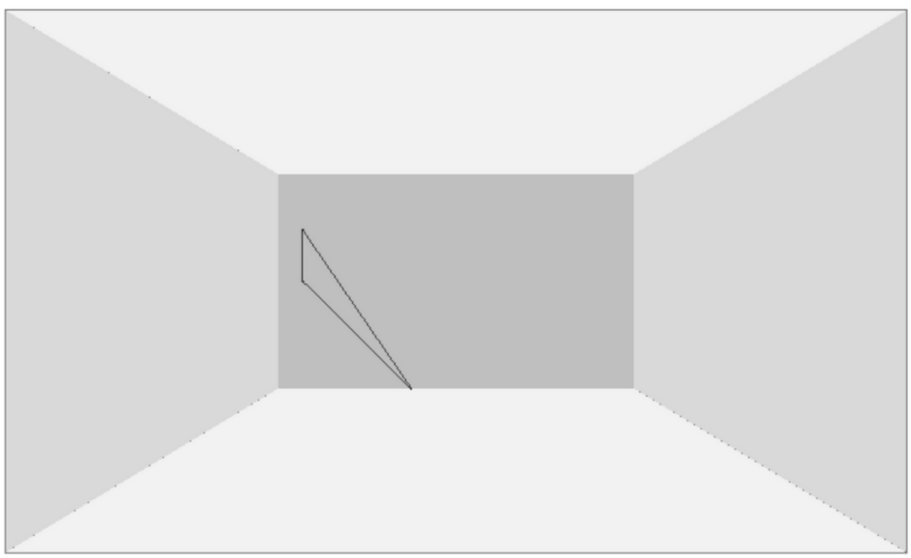


Figure 4.1: Passive Robot's Roadmap during the EnterExchange Stage

During the "EnterExchange" phase, Boxy3 is left by itself. Figure 4.2 illustrates the roadmap at this stage.

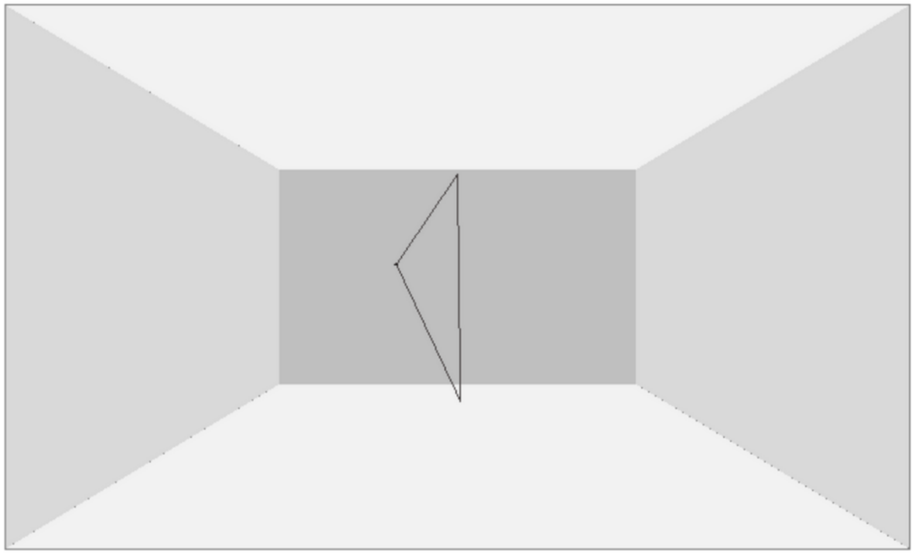


Figure 4.2: Boxy3 Robot's Roadmap during the EnterExchange Stage

Similar to Boxy3, Boxy2 is also left by itself during this stage. It's roadmap is shown in Figure 4.3.

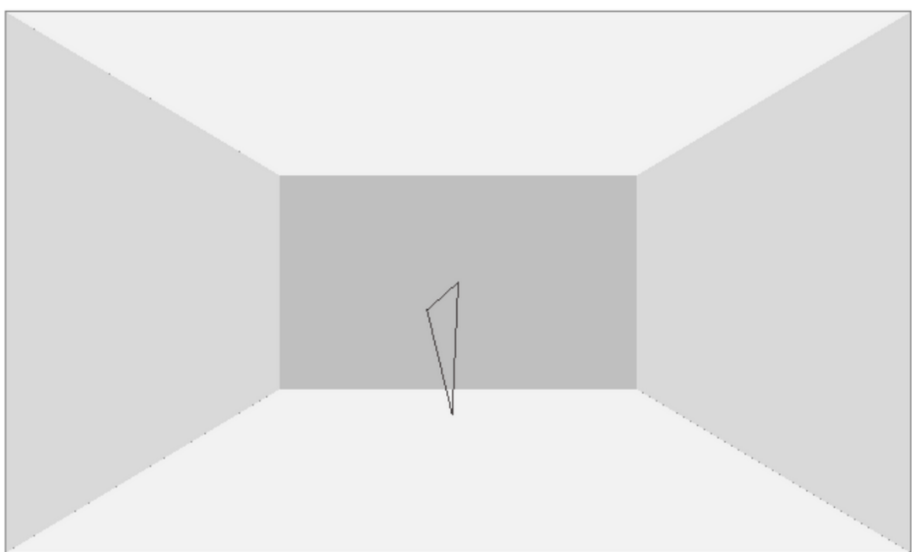


Figure 4.3: Boxy2 Robot's Roadmap during the EnterExchange Stage

Boxy1 is the robot that is currently handling the passive object at this phase in the handoff interaction. It’s roadmap is shown below in Figure 4.4.

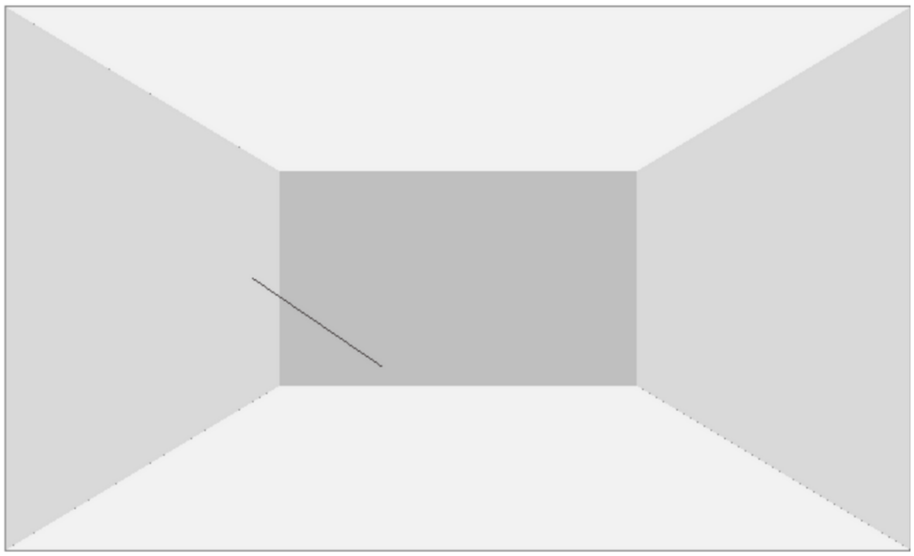


Figure 4.4: Boxy1 Robot’s Roadmap during the EnterExchange Stage

There are more roadmaps corresponding to each stage for each robot, but those were omitted for brevity. Currently, results only exist for situations with three or more robots and results for the dependent paths implementation have not yet been established.

5. CONCLUSION

A method for combining Interaction Templates and Reachable Volumes into a cohesive algorithm has been outlined in this paper. The advantages to this algorithm presented in this paper include its ability to work well for high degrees of freedom and constraints as well as operating efficiently for repeated operations such as handoffs or team-oriented robotics tasks. This is accomplished by leveraging the strengths of Reachable Volumes, namely its ability to precisely place the end-effector, during the sampling phase of Interaction Templates.

Progress has been made on the implementation of this algorithm for three or more robots and it has been shown that Interaction Templates can be made using this method for three or more robots using the Independent Sampling method. This allows for more complex tasks that may involve any number of robots. Additional progress has been made implementing Dependent Paths for three or more robots, however it is not yet at a presentable state. Further progress is needed on Dependent Paths for simple robots before any conclusions can be made about the efficiency of this algorithm.

6. FUTURE WORK

During the course of the project, the focus had to be reduced from the initial goal of working with manipulators with high degrees of freedom to working with simple robot designs in order to lay a foundation for the development of this algorithm. The current state of the project has only begun to finish the ground layer and has not progressed up to making this algorithm work for high degree of freedom manipulators and complex test cases. Furthermore, this iteration was done on a unique branch of the code that will take some time to transfer over similar functionality to other branches.

Future work would involve taking the work that has been done with Independent and Dependent sampling on simple robots and modifying it to work with high degrees of freedom manipulators. This could also take the form of modifying the existing high degree of freedom code to closer match the simpler examples. This may require the restructuring of some of the framework of the high degree of freedom manipulator code in order to accommodate some of the structures that were used to perform the Independent and Dependent Sampling. These structures include the "stages" and "groups" that are described in Section 3.2. Additionally, the implementation of Dependent Paths for three or more robots still needs to be finalized. There is a current version of the Dependent Paths implementation that is in development however it is not yet finished.

Once the Dependent Paths implementation is finished, the next step would be to test that implementation of the algorithm with the simple robot systems. This could take the form of a various battery of tests including large groups of robots, tight constraints on the handoff operation, using differently and oddly shaped robots, and the inclusion of obstacles in the environment. After this, runtime comparisons can be made between this algorithm and similar algorithms such as FFROb. This would provide a baseline for more simple operations.

The next step would be applying what was mentioned in the previous section to high degrees of freedom manipulators. After the code has been modified to reflect these changes, all of the same tests mentioned in the previous paragraph will be attempted, including larger groups, tighter constraints, different shapes, and obstacles. Additionally, testing should be done on increasing degree of freedom robots to show the strengths of this algorithm. Rigorous testing with large amounts of constraints or robots needs to be demonstrated in order to exemplify the strength of the algorithm as well as its versatility. After these tests, or any combination thereof, are performed, runtime analysis between this algorithm and other algorithms can be documented and the speed between these algorithms under a variety of constraints can be measured and compared.

REFERENCES

[1] J. Motes, R. Sandström, H. Lee, S. Thomas, and N. M. Amato, "Multi-robot task and motion planning with subtask dependencies," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2020.

[2] R. Sandstrom, D. Uwacu, J. Denny, and N. M. Amato, "Topology-guided roadmap construction with dynamic region sampling," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6161–6168, 2020.

[3] 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.

[4] C. Ekenna, *Improved Sampling Based Motion Planning Through Local Learning*. PhD thesis, Texas A&M University, College Station, Texas, 2016.

[5] M. Saha and P. Isto, "Multi-robot motion planning by incremental coordination," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5960–5963, 2006.

[6] T. McMahon, S. Thomas, and N. M. Amato, "Motion planning with reachable volumes," Tech. Rep. 13-001, Parasol Lab, Department of Computer Science, Texas A&M University, Jan. 2013.

[7] T. McMahon, S. Thomas, and N. M. Amato, "Sampling based motion planning with reachable volumes," Technical Report TR13-008, Department of Computer Science, Texas A&M University, College Station Tx., 2013.

[8] T. McMahon, S. Thomas, and N. M. Amato, "Reachable volume RRT," Technical Report TR14-007, Department of Computer Science, Texas A&M University, College Station Tx., 2014.

[9] X. Tang, S. Thomas, and N. M. Amato, "Effcient planning of spatially constrained robot using reachable distances," Tech. Rep. TR07-001, Parasol Lab, Dept. of Computer Science, Texas A&M University, Jan 2007.

[10] T. McMahon, S. Thomas, and N. M. Amato, "Sampling based motion planning with reachable volumes," Tech. Rep. TR13-008, Parasol Lab, Dept. of Computer Science, Texas A&M University, Sep 2013.

[11] T. McMahon, S. L. Thomas, and N. M. Amato, "Reachable volume RRT," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, (Seattle, WA, USA), pp. 2977–2984, May 2015.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] L. E. Kavraki, P. Švestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 566–580, Aug. 1996.

[18] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, vol. 22, pp. 560–570, Oct. 1979.

[19] R. Geraerts and M. H. Overmars, "Sampling techniques for probabilistic roadmap planners," pp. 600–609, 2004.

[20] G. Sanchez and J.-C. Latombe, "Using a prm planner to compare centralized and decoupled planning for multi-robot systems," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 2, pp. 2112–2119, 2002.

[21] J. E. Walter, J. L. Welch, and N. M. Amato, "Distributed reconfiguration of hexagonal metamorphic robots in two dimensions," in *Sensor Fusion and Decentralized Control in Robotic Systems III, Proceedings of SPIE* (G. T. McKee and P. S. Schenker, eds.), pp. 441–453, 2000.

[22] C. H. Yong and R. Miikkulainen, "Coevolution of role-based coordination in multi-agent systems," Tech. Rep. AI07-338, AI Lab, Dept. of Computer Science, University of Texas, Austin, TX, 2007.

[23] R. Gayle, W. Moss, M. C. Lin, and D. Manocha, "Multi-robot coordination using generalized social potential fields," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 106–113, 2009.

[24] P. A. O'Donnell and T. Lozano-Pérez, "Deadlock-free and collision-free coordination of two robot manipulators.," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 89, pp. 484–489, 1989.

[25] 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.

[26] e. a. Garrett, Caelan Reed, "Ffrob: An efficient heuristic for task and motion planning.," *Algorithmic Foundations of Robotics XI*, vol. 107, p. 179–95, 2015.