

Efficient Path Planning of Secondary Additive Manufacturing Operations

Keimargeo McQueen¹, Sara Darensbourg¹, Carl Moore Jr², Tarik Dickens³ and Clement Allen¹

¹Dept. of Computer and Information Sciences, Florida A&M University, Tallahassee, FL 32307, USA

²Dept. of Mechanical Engineering, FAMU-FSU College of Engineering, Tallahassee, FL 32310, USA

³Dept. of Industrial Engineering, FAMU-FSU College of Engineering, Tallahassee, FL 32310, USA

Abstract. We have designed a path planner for an additive manufacturing (AM) prototype that consists of two robotic arms which collaborate on a single part. Theoretically, with two nozzle equipped arms, a part can be 3D printed twice as fast. Moreover, equipping the second robot with a machining tool enables the completion of secondary operations like hole reaming or surface milling before the printing is finished. With two arms in the part space care must be taken to ensure that the arms collaborate intelligently; in particular, tasks must be planned so that the robots do not collide. This paper discusses the development of a robot path planner to efficiently print segments with two arms, while maintaining a safe distance between them. A solution to the travelling salesman problem, an optimal path planning problem, was used to successfully determine the robots path plans while a simple nozzle-to-nozzle distance calculation was added to represent avoiding robot-to-robot collisions. As a result, in simulation, the average part completion time was reduced by 45% over the single nozzle case. Importantly, the algorithm can theoretically be run on n-robots, so time reduction possibilities are large.

1 Introduction

Today's 3D printers are used to create end products for sale some even as critical and detailed as artificial limbs for children [1]. The time required to complete a part depends on many factors including but not limited to part size. Modern printers are faster than those just a few years ago, but when a person is new to 3D printing, they may be surprised by the amount of time needed to print what they consider a simple part. Moreover, after the part has finished being printed there often remains additional tasks, which we call secondary (or post) operations that must be completed to produce a usable and visually appealing part. A common secondary operation is hole refinishing where because of part shrinkage and printer resolution, holes are printed undersized and then drilled to the correct diameter afterward. These secondary tasks further reduce the time efficiency of the 3D part completion process.

This paper shows that the 3D part completion time can be reduced using DEXTER our AM prototype [2]. Named for its dexterity and modularity, DEXTER is a novel robot based system with dual Selective Compliance Assembly Robot Arms (SCARA) which can work on parts simultaneously as shown in Figure 1. Each arm has two links and three stepper motors to position the nozzle at the end of each arm in 3-DoF (degrees of freedom). The SCARA configuration is very appropriate for 3D printing where rigidity is desired along the z-axis and some compliance is desired in the x-y plane.

Unlike the h-frame rails that position the nozzle of a traditional 3D printer, the SCARA configuration allows more than one nozzle to travel in the part space without interference. With two nozzles, a part can be finished more quickly – twice as fast in theory.

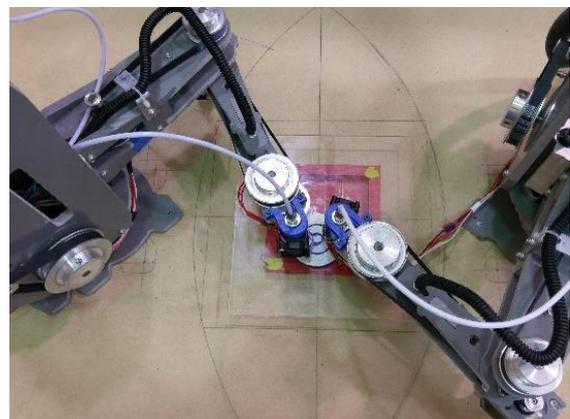


Figure 1. The DEXTER multipurpose hybrid machine is a dual SCARA manipulator shown here with nozzle equipped end effectors.

Furthermore, the second SCARA's nozzle can be replaced by traditional machine tools to enable the completion of secondary operations before the part has finished printing. Whether the second SCARA is used to print layer segments or perform secondary operations, the

motion of both robots must be coordinated for efficiency and to avoid collision with each other.

The problem of finding a time efficient path through all print segments (and secondary operations) is similar to the Traveling Salesman Problem (TSP) in computer science [3]. TSP has many variations, but the general problem description is as follows: given a finite number of cities where the cost to travel from any city to any other is known, find the minimum cost to travel through every city, starting and ending at the same city and traveling through every other city only once. In the simplest form, the costs are the pairwise Euclidian distances between cities.

TSP is a NP hard optimization problem, but there are many heuristic approaches discussed in the literature where useful results, while not always optimal, can be found in a fraction of the time required for optimal solutions [4]. One of the simplest heuristics approaches is called the nearest neighbor algorithm where the next city chosen is that which is nearest the current [5]. The nearest neighbor heuristic quickly solves the TSP resulting in a short tour that is generally not the optimal path.

Previous research has shown that path planning for 3D printer nozzles can be formulated as a TSP [6][7]. Moreover, it has been demonstrated that the nearest neighbor algorithm, where the nozzle always travels to the next nearest print segment, performs very nearly as well as more complicated heuristics [8]. Herein, we modify the nearest neighbor algorithm so that (a) it is applicable to the 3D printing context using multiple robots and (b) the robots' nozzles do not violate each other's proximity boundaries which could result collision. In reality, it is necessary that the robots' links also not collide, but this requirement has been tabled at this stage of the research so that we can focus on the applicability of the nearest neighbor solution to the path planning problem for a multi-nozzle based 3D printer like DEXTER.

2 Experimental Methodology

The multi-nozzle path planning algorithm was developed in MATLAB and operates as follows: every segment is made up of two cities - one at either end of a print segment. For a robot to travel to a segment it must satisfy two criteria: (1) the segment must be unvisited and (2) traveling to the segment and along the segment cannot cause the robot to violate another robot's proximity boundary. Of all the segments that satisfy these two requirements, the one closest to the robot's current position is chosen. If no segment can be reached without causing proximity violation(s) the choosing robot will remain stationary until the proximity violation(s) is removed. The proximity boundary is in effect a safety radius around each robot's nozzle. Once the robot arrives at the chosen segment it checks to see if motion to the other end of the segment is still safe. If so, then the robot prints the segment, but if the motion will cause collision, then the robot remains stationary until the violation is removed.

It would be too conservative to count as a proximity violation the intersection of two robot paths. Instead we consider whether at the given nozzle speeds the distance between two robots will remain more than the required

proximity value. To accomplish this, we simply need to know the robots' paths, their current positions along those paths, and their respective nozzle speeds and accelerations. All of this information is known in the 3D printer context. For simulation, the nozzle speed for each segment was randomly assigned from the following set of three: 1.0 mm/sec, 2.0 mm/sec or 3.0 mm/sec, and all of the robots travel between print segments at a speed of 6.0 mm/sec. The nozzle speeds are held constant for individual segments throughout the simulated runs as is the intersegment speed of 6.0 mm/sec. Lastly, the simulated nozzle accelerations were assumed infinite so that the desired nozzle speeds could be achieved in zero time.

3 Results and Discussion

Our simulated part layer contains 15 segments (or secondary operations) and 15 travel paths between these segments. The robot's start segment is randomly selected. An example solution to the single robot case is shown in Figure 2. The execution time ($n = 10$) averaged 231.37 +/- 2.32 seconds.

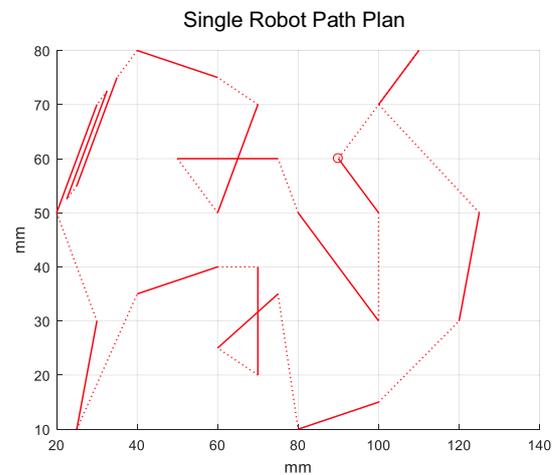


Figure 2. Example of path plan for one robot. Solid lines are print segments, and dotted lines are the intermediate paths. The robot started at the position indicated by a circle and immediately prints that segment before moving to one segment after the other according to the nearest neighbour heuristic.

To improve print time efficiency, the DEXTER system employs two robots. For two or more robots, the distance between them must be monitored to avoid collision; nevertheless, to compare the time to completion for two robots with the single robot case, we first ran the two robot simulation without regard to collision (proximity limit = 0.0 mm). The result is in Figure 3.

With two robots operating simultaneously and no proximity limit the print completion time ($n = 10$) averaged 122.91 +/- 1.57 sec. Therefore, using two robots reduces the print time required by almost 47% compared to a single robot. Even though the distance between the robots did not influence the path plan for this run, Figure 4 illustrates the distance between the robots as a function of time. It is noticeable, that without a proximity requirement the robots came within 2 or 3 mm of each other on two occasions.

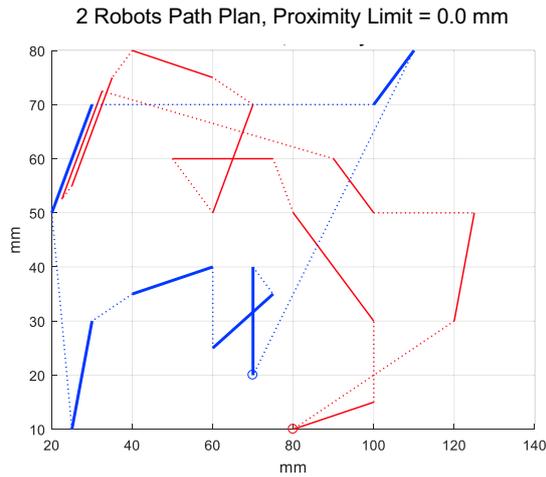


Figure 3. Path plan for two (2) robots without proximity limit. The thinnest line represents robot 1 print segments, and the thicker line represents robot 2 print segments. The robots start positions randomly chosen and indicated by circles.

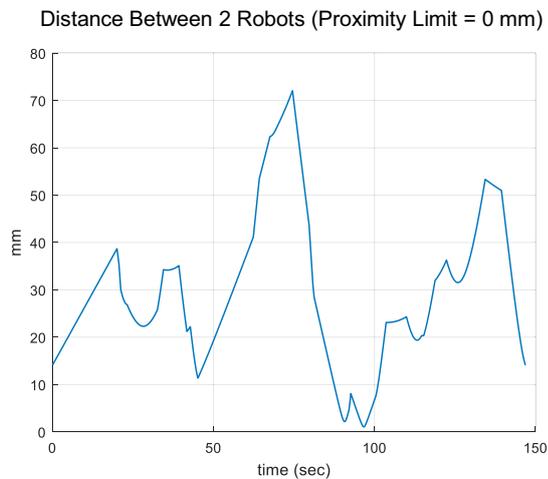
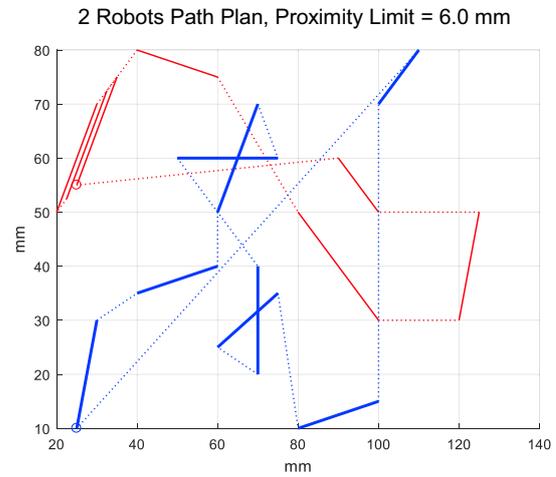


Figure 4. The distance maintained between robots 1 and 2 as they completed their print segments without proximity limit.

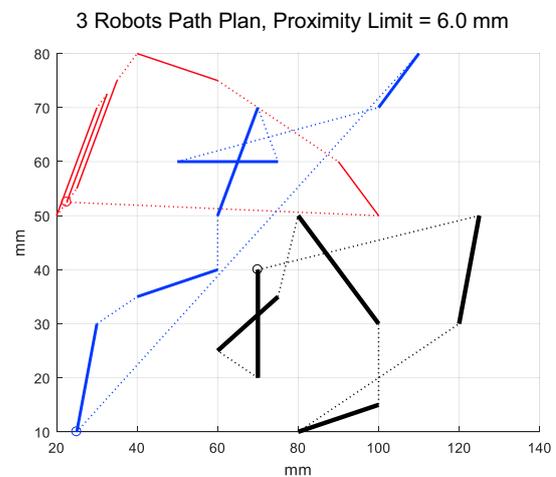
3D printing time can be reduced if more robots collaborate in the part space. If two robots can equally divide the printing of 15 identical segments (i.e.: 7.5 segments each), three robots could theoretically complete the same task by printing 5 segments each, representing a reduction in print time of 33%, or four robots by an additional 25%. But layer complexity and collision avoidance may preclude the equal division of the task such that the reduction in print time is not as great. To test this hypothesis, we ran simulations with 2, 3 and 4 robots operating simultaneously using proximity limits of 0.0 mm, 6.0 mm and 18.0 mm. Figure 5 shows representative path plans each with the 6.0 mm proximity limit.

For two robots and a 6.0 mm proximity limit, the path completion time increased about 4 seconds to 126.5 sec., still 45% less than with one robot. The paths chosen must deviate from potentially faster ones to maintain the minimum distance between the robots. But, increasing the number of robots makes up for the proximity limit increase resulting in time reductions not far from those

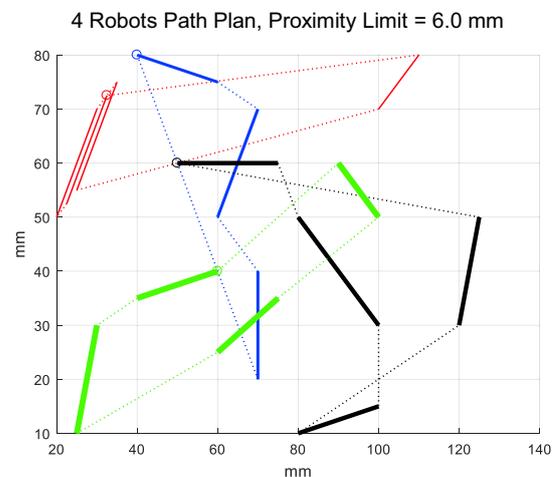
hypothesized above. Adding one robot for a total of three (Fig. 5b), reduced the completion time to 89.18 sec. or 30% less than the two robot case. Increasing to four robots (Fig. 5c), further reduced the path completion time to 72.31 sec. or an additional 19% reduction.



(a)



(b)



(c)

Figure 5. Path plans as follows: (a) two robots, (b) three robots, (c) four robots. Each plan executed with a proximity limit of 6.0 mm. Robot 1 print segments are plotted with the thinnest line; robot 2 print segments with a slighter thicker line, etc.

With a proximity requirement of 6 mm it is illustrative to compare runs with increasing numbers of robot to see how close the robots came to one another. Figure 6 demonstrates the calculated distances between all robot pairs on sample runs with 2, 3 and 4 robots.

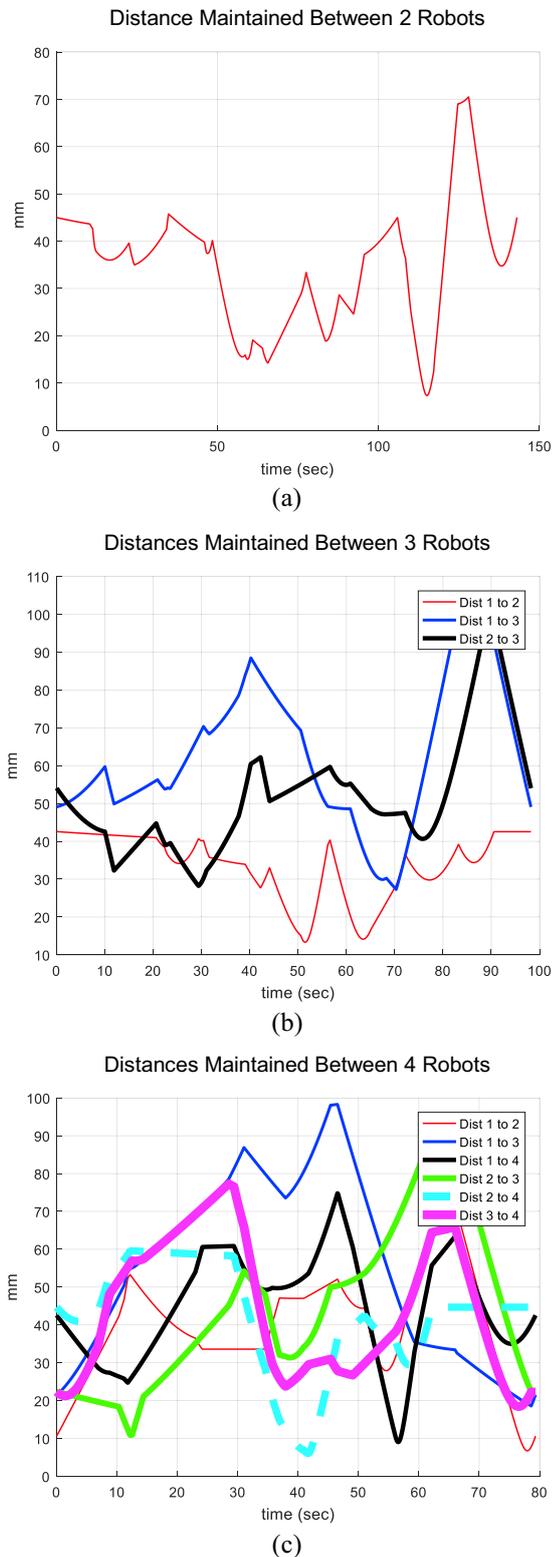


Figure 6. Calculated distances between multiple pairs of robots as follows: (a) two robots, (b) three robots, (c) four robots.

As can be seen in the Fig. 6, the robots never violated the proximity limit of 6.0 mm. However, the four robot case (Fig. 6c) produced multiple instances where the distance between a pair of robots was just outside the proximity limit. It was at these instances that one of the four robots had to reduce its speed to zero so as not to violate the proximity limit. Figure 7 illustrates the nozzle speed of each of the four robots during the run shown in Fig. 5c.

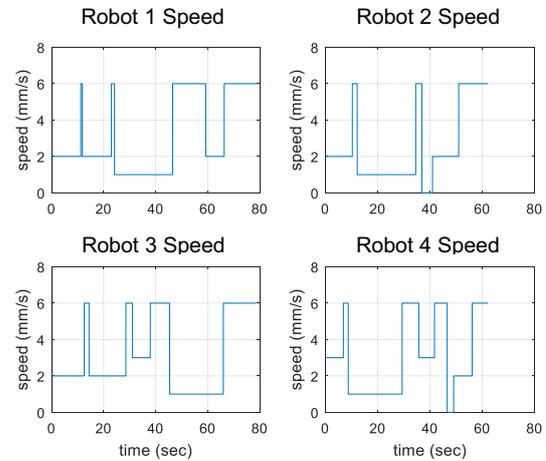


Figure 7. Four robot nozzle speeds for the path plan shown in Fig. 5c. The nozzle speeds were set for various print segments before the path planning begins and remained the same for all runs. Robots 2 and 4 reduced their speeds to 0 mm/sec during their tasks to avoid proximity violations, but completed their paths more quickly than robots 1 and 3.

Table 1 records the average amount of time required to complete the print segments for all runs including the 18 mm proximity case. Each simulation was done 12 times, and after the slowest and faster were removed, the remaining 10 were averaged. As the number of robots increased the print times decreased for all proximity limits. The standard deviation generally increased with increasing proximity. Larger proximity values increase the likelihood that one or more robots will have to pause during its task to clear a potential proximity violation. Also, for more than two robots, the increase in proximity limit does not significantly alter the time required for task completion.

Table 1. Average Task Completion Time.

Robots	Proximity (mm)		
	0.0	6.0	18
Avg. Task Completion Time (sec)			
2	122.91 ± 1.57	126.50 ± 2.58	131.92 ± 4.11
3	87.99 ± 1.37	89.18 ± 2.63	91.65 ± 5.65
4	69.58 ± 3.52	72.31 ± 6.65	72.75 ± 5.62

4 Conclusions and Future Work

Using the nearest neighbor heuristic to plan the nozzle paths of multiple 3D printing robotic arms can

significantly reduce the amount of time needed to print layer segments. For future work, collision avoidance of the robots' links will be added to the path planning algorithm and secondary operations like drilling and surface milling will be more explicitly investigated.

References

1. S. Smith, *Medical Humanities* **42**:259-264 (2016)
2. J.B. Frketic, S. Psulkowski, A. Sharp, T. Dickens, *Procedia Manufacturing*, **10**:1087-1096 (2017)
3. D. S. Johnson and L. A. McGeoch, *Local Search in Combinatorial Optimization*, vol. 1:215-310 (1997)
4. C. Rego, D. Gamboa, F. Glover, C. Osterman, *European Journal of Operational Research*, **211.3**:427-441 (2011)
5. S. G. E. Brucal and E. P. Dadios, *2017 IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, 1-6 (2017)
6. K. Fok, C. Cheng, C. K. Tse and N. Ganganath, *2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 382-385 (2016)
7. K. Fok, C. Cheng, N. Ganganath, H. H. Iu and C. K. Tse, *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1-5 (2018)
8. F. N. Ganganath, C. Cheng, K. Fok and C. K. Tse, *2016 2nd International Conference on Control, Automation and Robotics (ICCAR)*, 287-290 (2016)