

Developing Plans for Robotic Excavators

Sanjiv Singh¹

Abstract

We would like a robot excavator that is able to excavate a volume of soil according to specification. To this end we have developed a general method that can be used to provide prescriptions as to where and how to dig for a variety of excavator configurations and tasks. This paper presents simulation results of a front-end loader performing a loading task. In addition, preliminary results from a trenching experiments conducted with a robot in our lab are presented.

Introduction

Although excavation is ubiquitous in the construction industry, most day-to-day operations proceed on technology that is 30-40 years old— technology that has not kept pace with other industries. Even without the commercial incentive, automated earth moving machines are needed in worksites that are hazardous to humans. The most immediate application of autonomous excavation is to cleanup operations at sites where toxic and nuclear wastes have been stored. Many cleanup efforts are progressing slowly because they expose human workers to substantial risk. Another application is excavation on the Moon and Mars. NASA studies conclude that excavation will be one of the first activities necessary to establish planetary habitats. Since manned operation in space is extremely expensive, it is envisaged that robots will do much of the work necessary before habitats are fully established [NASA 90].

There has been some interest in remote controlled excavators for construction and hazardous waste site remediation [Kraft90][Muramatsu89]. The aim of this work is to remove the operator from the immediate workspace. All motions of the excavator, however, must be orchestrated by an operator who uses a master manipulator to instruct the excavator, typically by looking at a video monitor. Some researchers have investigated the execution of previously planned trajectories for an automated excavating robot [Bullock89][Bernhold90], and others have sought to develop gross plans

¹Graduate Assistant, Field Robotics Center, Carnegie Mellon University, Pittsburgh, PA. 15213.

for digging [Romero-Lois89][Apte92]. A few systems have shown greater autonomy [Whittaker 85][Seward92].

We would like a robot excavator that is able to excavate a volume of soil according to a specification. Excavation tasks range from loading from a pile of soil, to cutting a geometrically described volume of earth as is required in the digging of a trench or a foundation footing. Assuming that an autonomous excavating robot is equipped with a sensor that on-goingly provides shape of the terrain to be excavated, the one-dig problem may be formulated as follows: *Sense the terrain. Of all possible digs, consider a subset that the robot can feasibly perform and out of those, choose one that optimizes a cost criterion. Execute the dig.* Complete excavation is accomplished by concatenating a sequence of such plans. The problem can then be reduced to finding the best dig for a given terrain with a particular robot.

We propose that robotic tasks like excavation be represented in an *action* space, a space spanned by a compact set of parameters that encode the task. We will consider digging actions that are parameterized by a small set of variables. The values that these variables can attain define a multi-dimensional space of possible actions. In this space, every point represents a unique digging plan and we can partition the space to exclude those plans that are not feasible. Certain actions are excluded because they are not feasible. For example, a dig may not be feasible if a robot is required to reach out side its workspace, or, if the dig intrudes past specified geometric boundaries or, if the force required to make a particular dig is greater than that the robot can develop. Among the set that satisfies all the constraints, it remains to find one dig that optimizes a cost function, for example, the amount of soil excavated.

There are two advantages of formulating the problem such that the action space is independent of the configuration of the digging machine. First, it provides generality. Consideration of a variety of machines and terrains results in different constraint surfaces on the parameter space, but the space itself does not change. Second, digs parameterized as a combination of task variables are easier to analyze than those that are represented as trajectories in the joint space of the robot actuator.

An Example

In order to provide an intuitive understanding of the proposed approach, let us consider an extended example in a two-dimensional world. Figure 1 shows a terrain that must be excavated and Figure 3 shows the mechanism that is to be used.

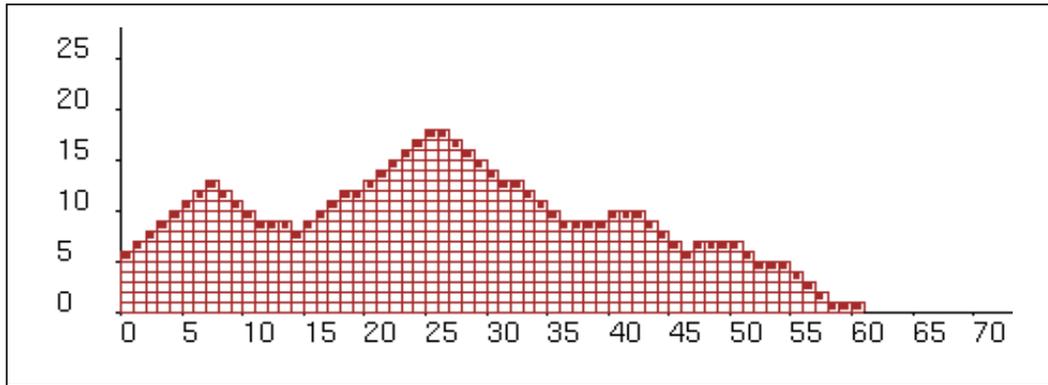


Figure 1. The terrain to be excavated

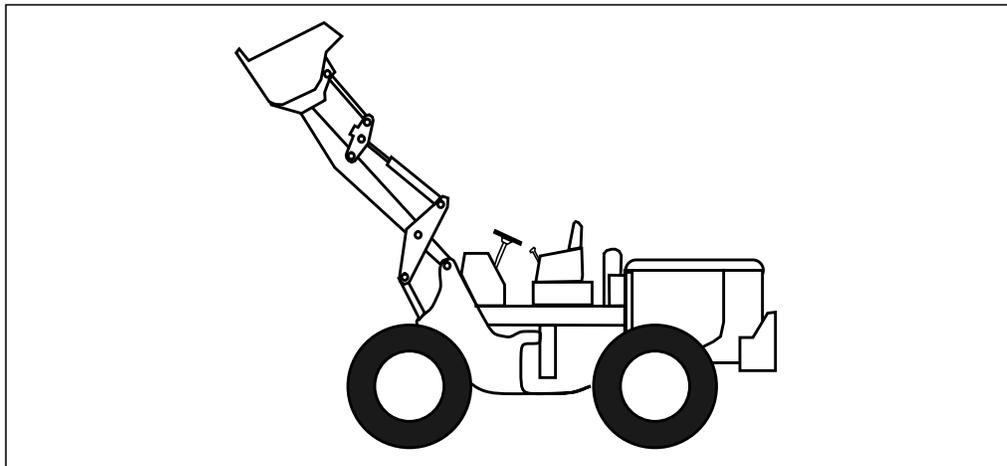


Figure 2. Mechanism to be used for excavation.

This sort of device is commonly called a “bucket loader” or a “front-end loader” and can be automated. The loader is to completely excavate the pile, without intruding below the surface of the ground. In this example, we show how geometric and force constraints are imposed on the action space for excavation. Let us represent the action space by a three parameter set (α, d, h) . α is the angle at which the excavator bucket goes through the soil along a line of length d . h is the height at which the bucket enters the pile as shown in Figure 3.

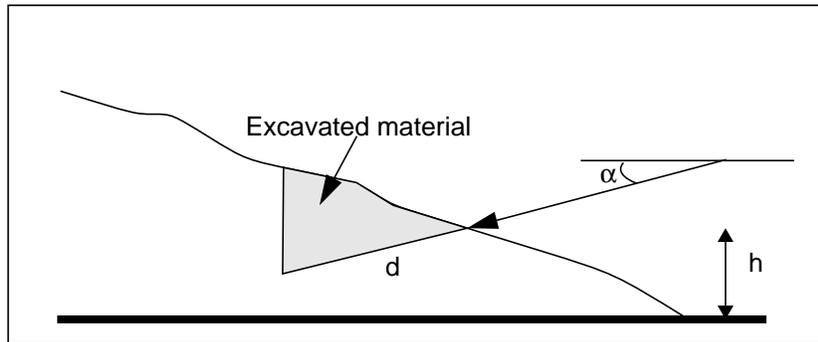


Figure 3. A prototypical dig.

Let us consider three types of geometric constraints that can be imposed on the action space: *reachability*, *volume* and *shaping*.

Reachability Constraint: Every candidate dig can be mapped to a trajectory for the bucket tip to follow, and a standard inverse kinematics method can be used to find the corresponding joint displacements. A candidate dig may fail this constraint if it requires the excavator reach outside its workspace (exceed its joint limits) or if in the course of the dig, one or more of the links are required to interpenetrate the terrain. Modeling the excavator (Figure 2) as a P-R-R manipulator, the composite constraint surface due to the terrain and mechanism above is shown in Figure 4. The surface represents the boundary between the reachable and nonreachable digs— all points below the surface represent digs that meet the reachability constraint.

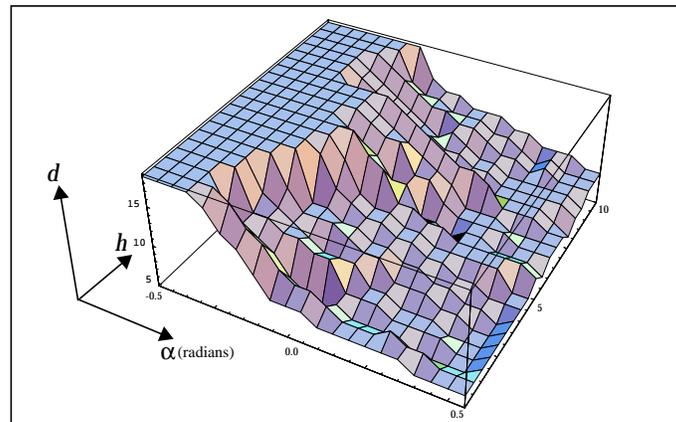


Figure 4. The reachability constraint.

Volume Constraint: Since the excavator bucket can only hold a volume V_{\max} , a digging motion triplet should not excavate more than this amount of soil. This gives us a

further basis on which we can limit the set of feasible digs. This constraint is shown in Figure 5.

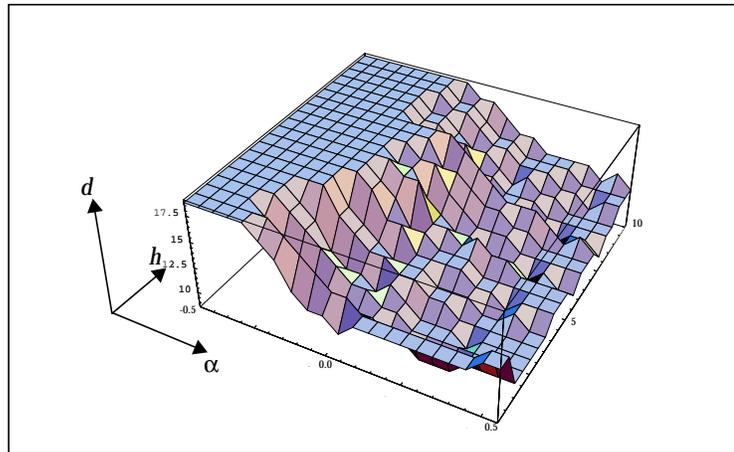


Figure 5. The volume constraint.

Shaping Constraint: This constraint is given by the goal state of the terrain. In general this may be an arbitrary, stable, geometric specification of the earth. For our example, we will limit digs so that they do not intrude below the surface of the ground. The resulting constraint surface is shown in Figure 6.

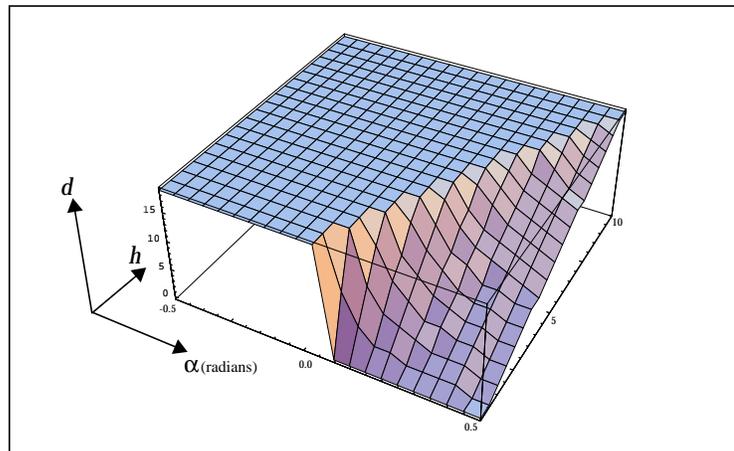


Figure 6. The shaping constraint.

If we assume that the robot excavator is infinitely strong— that it can muster any torque required, then the type of constraints discussed above are sufficient. More realistically, for robots with torque limits, we must consider the forces required to accomplish digging. Using an analysis of shear surfaces developed during excavation, the action space can now be further constrained based on whether or not it is possible for

the robot to generate the required forces for perform a candidate dig for example as shown in Figure 7.

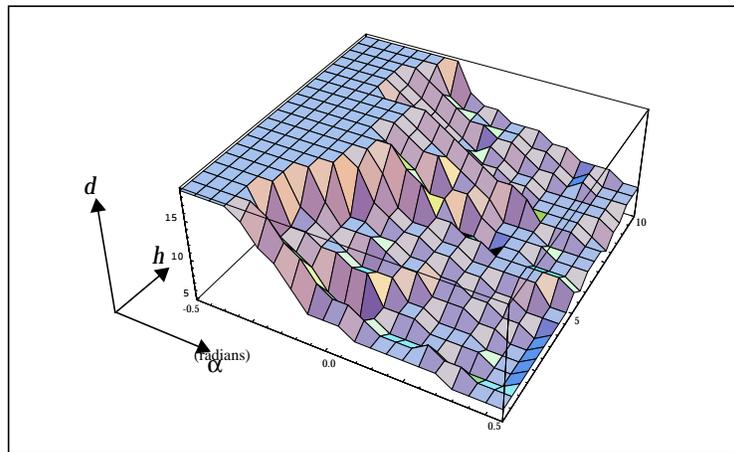


Figure 7. The force constraint.

Search for an “Optimal” Dig

Figure 8 shows the intersection all the constraints (geometric and force) discussed above.

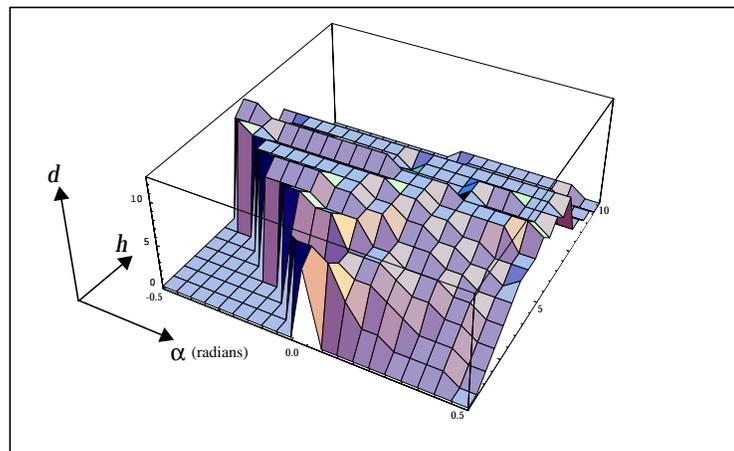


Figure 8. Intersection of geometric and force constraints.

This space² can be searched using a number of methods to find the optimal triplet that will provide the maximum benefit— in this case the amount of soil excavated. If the action space is small (represented by a few parameters) it may be possible to enumerate all the options by discretizing the space. Otherwise, as is often the case for describing realistic digs (larger number of parameters required), a numerical method is

²Note that this composite space changes (because the constraints change) after every dig and the optimization procedure must be executed to select each dig.

necessary. Since there is no guarantee of a unique extremum in the cost function, a method like simulated annealing can be used to optimize the cost function [Singh 91]. Once the optimization procedure has selected a dig, it can be mapped back to the joints of the robot and is guaranteed to succeed modulo the correctness of our models. It is not surprising that the one-step plan suffers from problems common to "greedy" methods; it selects the dig that optimizes the cost function at the next step only, and in general, will generate a sequence that is suboptimal. Nevertheless, our experiments have shown a greedy method to provide acceptable results.

Simulation Results

Figure 9 shows an example of excavation using a loader under simulation. Skeleton links of the excavator are shown in the process of digging.

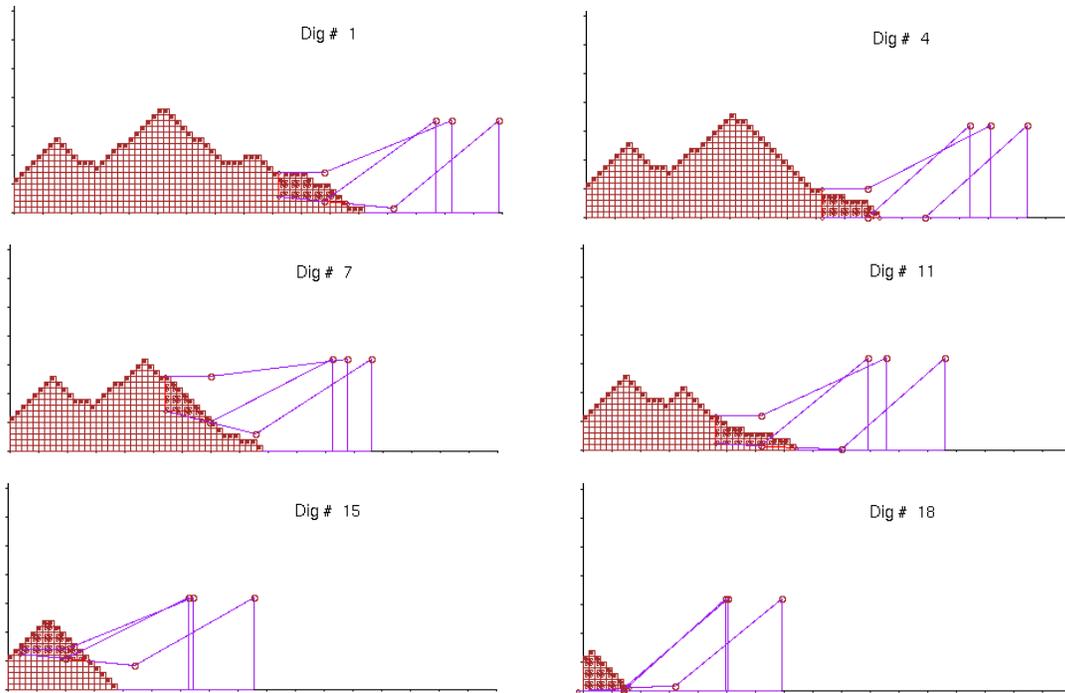


Figure 9. Simulated progression of a loading task.

We have used the same method described above to produce digging plans for a very different mechanism. The task is to dig a trench to a specified depth. Our experiments were conducted in a testbed that consists of a sandbox (2.5m x 2.5m x 1m), a Cincin-

nati Milacron T3 hydraulic robot outfitted with an excavator bucket and a Perceptron laser range finder. The setup of our testbed is shown in Figure 10.

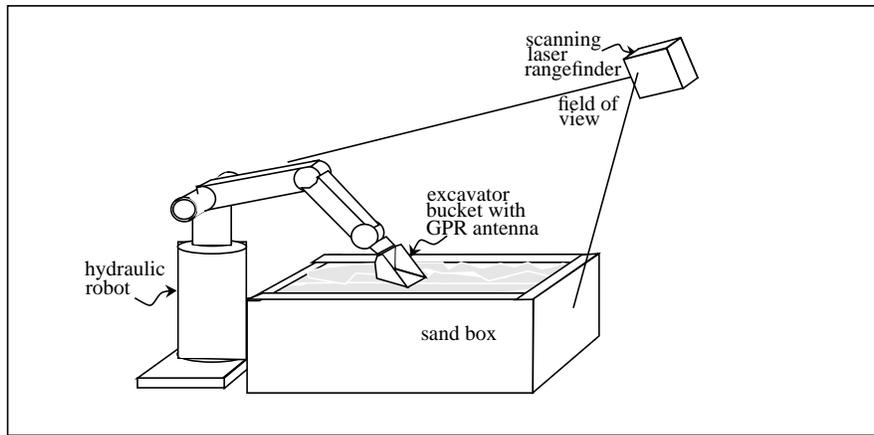


Figure 10. Excavation testbed

In a recent experiment, the robot was required to produce a trench 30 cm deep, 20 cm wide and 50 cm long. To produce such a free volume, extra soil must be excavated because the soil in our testbed consists of dry, fine sand. Hence the walls of the trench collapse until they are angled at less than the natural angle of repose of the soil. The planner continues to prescribe digs until the specified volume has been completely excavated. Figure 11 shows a cross-section of the sandbox along the long axis of the trench, as the task progresses. The best dig selected at each step is shown as a triangle that connects the points where the bucket enters the terrain, the point to which it digs to and then the point to which the bucket curls.

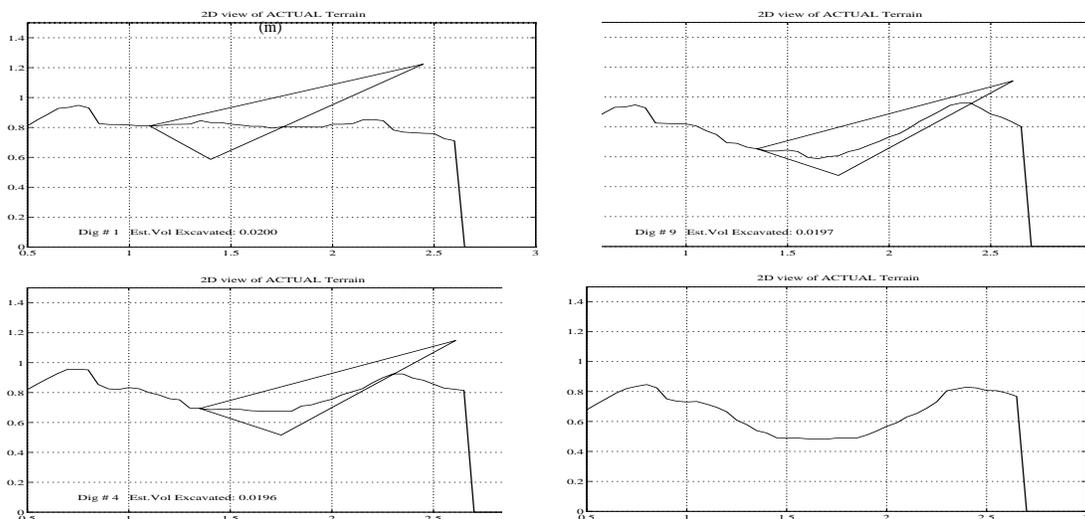


Figure 11. Progression of terrain during trenching task.

Summary

We have proposed a method to plan excavation tasks. The planning method requires prior parameterization of a task to reduce consideration of candidate plans to a manageable set. Such an abstraction provides an increase in efficiency since it keeps us from having to consider the infinite set of all possible paths that the excavator bucket might travel through. While such a scheme is harder to program (mainly because of difficulties in modeling and calibration) than a more reactive scheme, it is easier to analyze and generalize. We suggest that such a systematic method is useful because it allows researchers to learn about the problem—to distinguish between significant and negligible effects. Hence the approach is not only a means to a solution for robotic excavation, but also a means of analysis and learning about how to represent the task.

References

- [Apte92] Apte, S., “A Representation for Spatial Constraints in Material Removal,” Ph.D. thesis, Civil Engineering Department, Carnegie-Mellon Univ., July 1992.
- [Bernhold90] Bernhold, L., “Motion and path Control for Robotic Excavation,” submitted to ACSE Journal of Aerospace Engineering, Sep 1990.
- [Bullock89] Bullock, D M., Oppenheim, I. J., “A Laboratory Study of Force-Cognitive Excavation”, in *Proc. Six International Symposium on Automation and Robotics in Construction*, June 1989.
- [Church81] Church, H. K., *Excavation Handbook*, McGraw Hill, 1981.
- [Kraft90] “Force Feedback Excavator and Material Handling System,” Kraft TeleRobotics, Inc. Overland Park, Kansas, July 1990.
- [Muramatsu89] Muramatsu, T., Kenji, C., “A Concept of Control System For Construction Robot,” in *Proc. IEEE/RSJ International Workshop on Intelligent Robots and Systems*, September, 1989.
- [NASA90] Proceedings of FY89 Workshop on Extraterrestrial Mining and Construction, Lyndon B. Johnson Space Center, LESC-27585, 1990.
- [Romero-Lois89] Romero-Lois, H., Hendrickson, C., Oppenheim, I., “A Strategic Planner for Robot Excavation,” in *Proc. Sixth International Symposium on Automation and Robotics in Construction*, June 1989.
- [Seward92] Seward, D., Bradley, D., Mann, J., Goodwin, M., “Controlling an Intelligent Excavator for Autonomous Digging in Difficult Ground”, in *Proc. Ninth International Symposium on Automation and Robotics in Construction*, June 1992.
- [Singh91] Singh, S., “An Operation Space Approach to Robotic Excavation,” in *Proc. IEEE Symposium on Intelligent Control*, Alexandria, August 1991.
- [Singh92] Singh, S., Simmons, R., “Task Planning for Robotic Excavation,” In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Raleigh, 1992.
- [Whittaker 85]Whittaker, W., Turkiyyah, G., Bitz, F., Balash, J., Guzikowski, R., Montgomery, B., Akdogan, R., and Swetz, S., “First Results in Automated Pipe Excavation,” in *Proc. of the Second International Conference on Robotics in Construction*, Pittsburgh, May 1985.