A Practical Comparison of Motion Planning Techniques for Robotic Legs in Environments with Obstacles

Daniel Chavez-Clemente
Department of Aeronautics and Astronautics
Stanford University
Stanford, CA 94305–4035
Email: dchavez@stanford.edu

Tristan B. Smith
Intelligent Systems Division
NASA Ames Research Center
Moffett Field, CA 94035–1000
Email: tristan.b.smith@nasa.gov

***Abstract*—We present a practical comparison of three different methods to plan single-step motion for the six-legged ATHLETE robot.**

I. Introduction

ATHLETE is a large six-legged tele-operated robot designed for lunar operations[1]. Each foot is a wheel; travel can be achieved by walking, rolling, or some combination of the two. Operators control ATHLETE by selecting parameterized commands from a command dictionary. While rolling can be done efficiently with a single command, any motion involving steps is cumbersome - even a single step includes multiple individual commands (for example, 'lift leg 3 by 40 cm').

Our goal is to improve operator efficiency by automatically generating sequences of motion commands that move a foot to a new position. While a standard robotics approach is able to quickly solve most instances, ATHLETE operators describe the resulting motion as unpredictable and non-intuitive. We show that two versions of A* search can achieve qualitatively different results, and consider the advantages and disadvantages of the three approaches.

II. Three Approaches

Our first approach is Single-query Bi-directional planner with Lazy collision checking (SBL)[2]. Beginning with a graph consisting of the start and goal configurations, it randomly samples configurations and checks whether they are feasible and can be connected without collisions.

Our second approach uses A* search through ATHLETE's task-space. We discretize the three-dimensional space around ATHLETE into 10 cm intervals and consider ways to move the foot through that grid, with each position able to reach the six immediately adjacent to it.

Our third approach uses the same A* algorithm, but in configuration space, resulting in a branching factor of twelve.

III. Results

We consider two very different sources of feedback concerning the quality of the three approaches. First, we consider hundreds of experiments comparing the approaches on real and randomly-created terrain; for each run, we consider the running time of an algorithm, whether or not it was able to find a solution, and the quality of the resulting solution. Second, we consider the less-scientific, but arguably more valuable, feedback of ATHLETE operators.

Table I summarizes the results by showing how the three approaches rank for the various qualities the end user would like to see in a solution. The SBL approach is superior in terms of running time and success on difficult instances. However, ATHLETE operators prefer solutions generated by the A* approaches, partly due to a preference for a completely deterministic approach, and partly because the resulting motions appear more natural when observed.

TABLE I
Summary of Results: Ranking the Approaches

Desired Qualities	SBL	Task A*	Config A*
Average Running Time	1	2	3
Worst-Case Running Time	1	2	3
Success on Difficult Instances	1	3	2
User Comfort With Approach	3	1	2
User-Determined Quality	3	2	1

IV. Conclusion

Due to ATHLETE's size (almost 1 ton) and intended operating environment (near astronauts and far from the ability to easily repair damage), safety is critical and operators may lean towards the familiarity and comfort of an A* approach. On the other hand, they want an approach that is fast and able to solve the most difficult instances (when they will need it the most). For now, all three approaches are made available to ATHLETE operators; in future however, we hope to produce a hybrid algorithm that achieves the best of all worlds.

References

[1] Brian H. Wilcox et al., ATHLETE: A cargo handling and manipulation robot for the moon, *Journal of Field Robotics*, 24(5), pp. 421–434, 2007.
[2] Gildardo Snchez and Jean-Claude Latombe, A Single-Query Bi-Directional Probabilistic Roadmap Planner with Lazy Collision Checking. R.A. Jarvis and A. Zemlinsky (Eds): *Robotics Research*, STAR 6, pp. 403–407, 2003