Integrating Simulated Tensegrity Models with Efficient Motion Planning for Planetary Navigation

Zakary Littlefield, Ken Caluwaerts, Jonathan Bruce, Vytas SunSpiral, Kostas E. Bekris

***Abstract*— Tensegrity-based robots use compression elements and tension cables to create lightweight structures that can reconfigure their shape. These characteristics are especially suited for planetary exploration, including for hard to traverse areas, such as lava tubes. While these capabilities are desirable for transporting these robots beyond Earth as well as reducing material costs, they complicate the control process. With such dynamic and reconfiguring parts, both simulating the motions of the tensegrity robot and planning future motions become challenging. New simulation tools for tensegrity rovers and state-of-the-art planning algorithms have been recently developed which can help to address these challenges, but have yet to used in tandem. This work integrates a recent sampling-based motion planner, which has been shown to converge to asymptotically optimal solutions even for systems with dynamics, with a novel tensegrity rover simulation tool, which has been verified in terms of accuracy against a hardware prototype. This paper shows that it is possible to get complex, long-duration trajectories for planetary navigation through this integration. At the same time, this integration is computationally demanding which motivated a parallel implementation of the proposed integration. With the parallel implementation, it is possible to observe improving path quality as computation time increases. This framework allows the consideration of planning under uncertainty to compute robust solutions, which is even more computationally demanding.**

I. Introduction

Tensegrity-based structures have been proposed as flexible robotic systems [1], [2]. They provide compliance and load-sharing, which allow for dynamic maneuvers and reconfiguration over difficult terrains while maintaining structural integrity. In addition, it is possible to change the shape of the robot to allow for movement in previously unexplored areas, such as caves and lava tubes. Nevertheless, controlling tensegrity robots is challenging relative to other robot classes. There has been exciting progress on providing locally valid gaits [3], in some cases through the use of central pattern generators [1], [4], and has been evaluated on physical robots (see Fig. 1). These breakthroughs allow moving the robot in a desired direction in limited scenarios. It is harder, however, to achieve long-term navigation or reconfiguration.

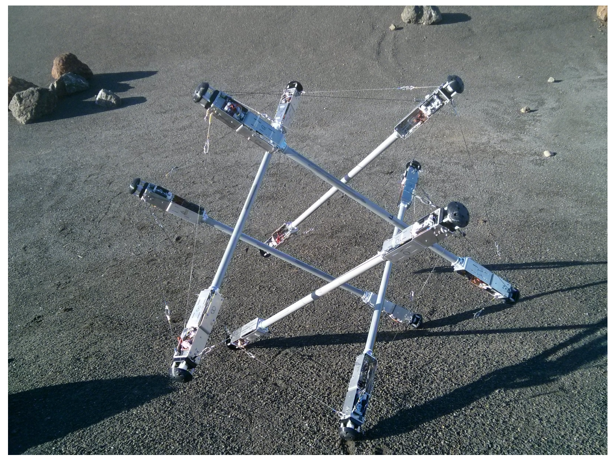

Fig. 1. SUPERball prototype from NASA Ames Research Center [3].

The generation of purposeful motions requires global planners, which reason over long horizons, consider terrain complexity, and provide diverse paths for science teams. Such methods have to deal with the high-dimensionality of the robot, the effects of contacts with the ground on the robot's dynamics, and noisy actuation. Rover navigation typically looks to search-based methods for providing long-term motion plans (see [5] for one example of such a method). It is helpful for these methods if the rover is as simple as possible, or can be simulated with simplified models. In addition, these methods rely on state or control discretizations, potentially disregarding solutions. For more sophisticated robots, such as tensegrity robots, these methods may not scale well.

A promising solution to this planning problem involves using sampling-based motion planners [6], [7], which have been shown to be successful when dealing with high-dimensional robots. Most sampling-based methods also can also use simulation tools as models of the robot. These sampling-based methods can also achieve asymptotic optimality under certain conditions [8]. The asymptotic optimality property states that given enough computation time, these sampling-based algorithms can return the optimal solution with probability approaching one. It also means that the solutions returned by these algorithms practically improve as more computation is provided. Until recently, these desirable properties could not be achieved in the case of highly dynamical systems.

A more recent development is an algorithm that provides asymptotic optimality for systems with dynamics [9]. While the theoretical properties require that the robot conform to certain smoothness assumptions, good performance is achieved in other cases as well. Using this method, finding paths of increasing quality for systems with dynamics or physically-simulated systems is now possible in a reasonable amount of time. In addition, this method can operate while planning under uncertainty by using a particle representation to model multimodal belief distributions and nonlinear dynamics [10]. As a sampling-based planner, it is possible to use a physics simulator as the robot model, but in order for the motion plans computed to be useful, the simulator must accurately model the robot to reasonable degree.

For simulating the high-dimensional tensegrity robots, a new software tool called the NASA Tensegrity Robotics Toolkit (NTRT) has become available to simulate tensegrity robots through the use of a physics engine [11]. Such simulations require significant computational resources due to the complex dynamics and contacts (tension cables, terrain contacts, shared force loads). The benefit of this expensive simulation is that it is shown to accurately approximate the real-world prototypes [3].

II. Proposed Integration

This work integrates the NTRT simulator [11] with the recent sampling-based framework [9] to perform long-horizon planning for tensegrity robots. To the best of the authors knowledge, this is the first time it has become possible to plan for tensegrity robots while taking dynamics into account, i.e., not just in a quasi-static manner as in [12].

A. Simulator and Robot

The tensegrity evaluated in the planning method is the SUPERball [3], which is a prototype robotic platform being built at NASA Ames Research Center. This structure has six rigid components arranged to mimic a icosohedron shape. This rigid elements are modeled as dynamic rigid bodies with 12 degrees of freedom each (three translational components, three rotational components, and their corresponding velocity terms). Movement is achieved by contracting the cables that connect the rigid elements. These contractions create forces on the rigid elements that cause the entire structure to reconfigure. Given enough change in the structure, rolling will occur, thus achieving locomotion.

The physical prototype at Ames is currently rigged with 12 actuated cables and 12 passive (not actuated) cables. An actuated cable is one that has motors that can actively contract the cable and release the tension of the cable. A passive cable maintains a maximum length, but can contract freely. This limitation for actuated cables was mainly due to the original prototype design having limitations for each rod. Future prototypes will have all cables actuated, thus providing more movement capabilities. The integration results presented here use 24 actuated cables.

B. Algorithm

The high level planning methodology is derived from recent work on Stable Sparse-RRT (`SST`) [9]. `SST` works in a similar fashion to other sampling-based tree planners, in that it follows a basic structure. First, a randomly sampled state is generated. Then, this node helps select which existing tree node is going to be expanded. The expansion step, otherwise known as a propagation, creates new nodes in the tree. `SST` has an intelligent selection of tree nodes and only adds nodes that have good path quality relative to other nodes nearby. These modifications allow for `SST` to achieve path quality guarantees, even for systems with dynamics. For more details about the algorithm, see [9], [10], [13].

One advantage that `SST` has relative to other sampling-based planners that can achieve asymptotic optimality is that `SST` only requires a forward propagation module versus the other that require a steering module. The main difference between these components is that a forward propagation only requires a start state while steering requires both a start and an end state. In the case of tensegrity-robots, the constant interaction with the environment's terrain and obstacles that impact the robot's dynamics make a steering function impossible to develop. On the other hand, NTRT provides all the necessary components for forward propagation, making `SST` uniquely tailored to the task of finding high quality motion paths for tensegrity robots.

C. Implementation Details

In order to improve the efficiency of the planner, a parallel implementation was used. Because the forward propagation for the tensegrity robot is costly relative to the other algorithm steps, if many of these propagations are done in parallel, the overall performance of the planner improves. A substantial improvement is observed in the number of iterations that can be performed in a given amount of time. In Figure 2, the number of iterations executed in two minutes is plotted versus the number of parallel propagation nodes. For the experimental evaluation, 20 parallel nodes are used.

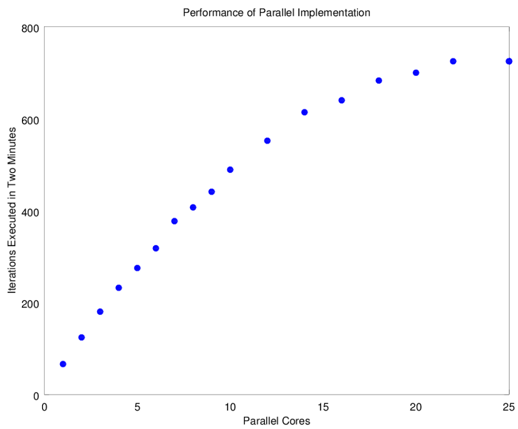

Fig. 2. Algorithmic performance for different number of parallel nodes. All runs ran for two minutes, and all nodes are on the same machine with 32 execution cores.

When moving to planning under uncertainty, the correct representation of uncertainty must be chosen. In many other domains, a Gaussian distribution is chosen, but is not appropriate for highly dynamical systems, such as tensegrity robots. This is due to their nonlinear behavior that likely will cause the uncertainty to follow multi-modal distributions, i.e. have multiple probability peaks rather than one. For this reason, a particle-based representation is chosen, where a set of particles approximate the underlying probability distribution.

Because a particle representation is used when planning under uncertainty, the computational cost of planning is increased significantly. Each particle must be simulated independently of the other particles, meaning NTRT must be called for each particle. Since this simulation is the dominant computation even when planning without uncertainty, improvements need to be made to make simulations faster. By taking advantage of the independence of the particles, a parallel extension is performed, where multiple particles can be extended at the same time, similar to the original setup. Instead of having propagations happen from anywhere in the search tree, many propagations are performed from the same start set. Then, all resulting states are composed together into one updated belief distribution.

III. Evaluation

The integration of NTRT with a sampling-based planner requires significant computational resources. This is mostly due to a basic primitive that a sampling-based planner requires, the forward propagation primitive. This forward propagation primitive in most cases is fast, but is a computational bottleneck when a physics engine is used. This is the case when using NTRT and this influences the planning execution time.

In this section, different scenarios are constructed where the tensegrity robot must traverse from its start position to a goal region. A simple problem is shown first, which is only the task of moving from the start to the goal. Then, more complex scenarios are introduced where terrain and obstacles are introduced. Terrain affects the basic movement of the robot, while the obstacles impose hard constraints on the robot that require reconfiguring to traverse around. Finally, some of the difficulties relating to planning under uncertainty are presented.

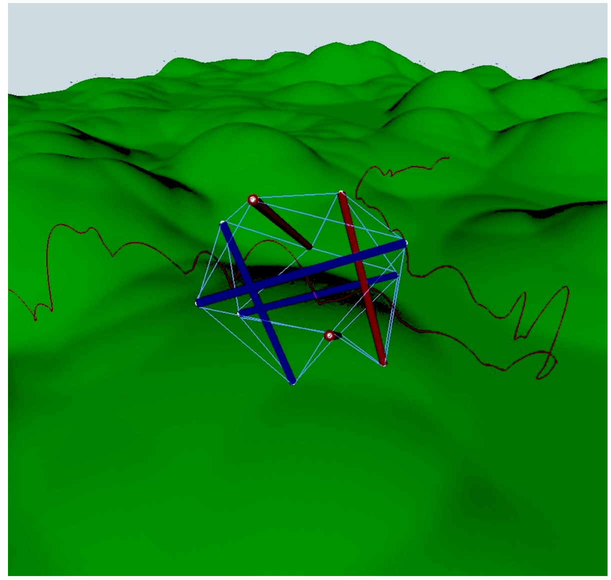

Fig. 3. An example path for the SUPERball tensegrity robot. This path also considers the terrain effects through the physics simulator.

A. Traversal Planning

For an initial test, a plan for moving without invalid regions is performed. The best path in this setup is as close to a straight-line between the start and goal as possible. This is not directly achievable given the dynamics of the robot. An example planned trajectory that considers terrain is shown in Fig. 3.

An example path planning tree is shown in Fig. 4 where the paths shown are for the center of a rod in the structure. The goal for the robot is the top right of the figure. The tree illustrates the inherent dynamics of the

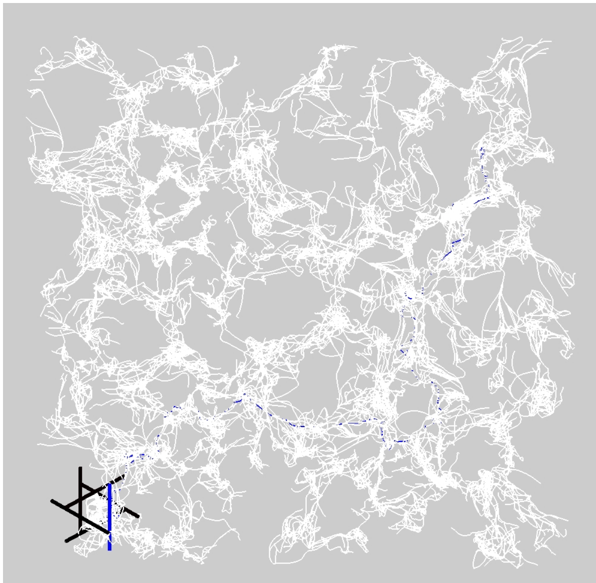

Fig. 4. An example tree computed from the motion planner. This example has no invalid regions.

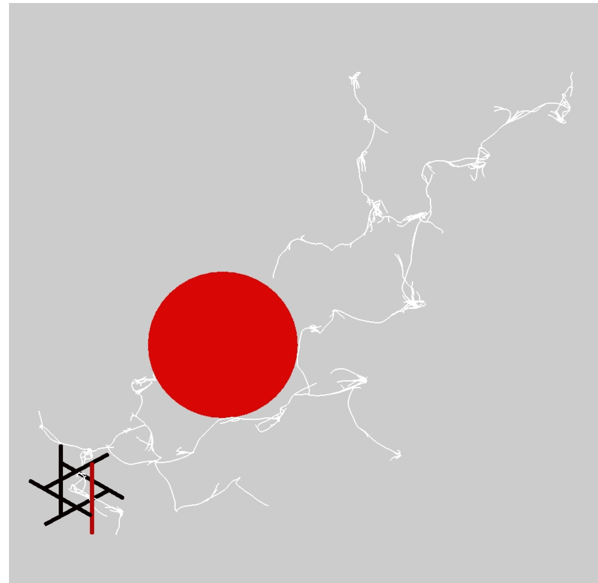

Fig. 5. An example tree computed from the motion planner. This tree has to avoid the red region, which causes the robot to traverse around it. Invalid regions could correspond to craters or inescapable areas.

SUPERball and how moving in straight lines is difficult even on flat terrain.

B. Terrain and Obstacles

In order to get closer to real mission objectives, tests were conducted with altered terrain and obstacles. The task to solve here is still to traverse to a goal region, but the terrain causes new behavior to occur. The effect of gravity influences the motion of the robot going up hills, but can also propel the robot down slopes.

Another setup that is shown here is a traversal around obstacles. These obstacles can represent the irregular topography of a cave-like environment. This highlights the need for high-level planners, since other methods may not be able to effectively reconfigure to achieve motion through tunnels. An illustration of the motion planner's tree is shown in Fig. 5.

C. Planning Under Uncertainty

A trajectory computed in the belief space is shown in Fig. 6. Due to noise in actuation, different final states states may be reached, which composes a belief over the actual state of the robot, illustrated as transparent shapes of the robot. Planning in belief space has higher computational cost relative to state space planning, but provides the benefit of robustness to errors.

Another interesting property that was discovered while planning under uncertainty is that the SUPERBall can inherently reduce its uncertainty with specific motions. This behavior arises due to the different faces that

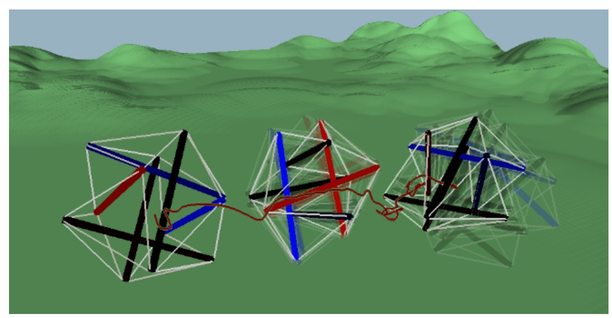

Fig. 6. An example trajectory computed when planning under uncertainty. The transparent versions of the SUPERball show different possible futures given uncertain actuation.

can be touching the ground at any given time. Even with small errors in actuation, a similar resting state can be achieved by not changing control inputs too rapidly. In addition, the set of particles quickly diverges and clusters into multiple modes (see Fig. 7 for an example). It may be possible to exploit this behavior in an intelligent way to help reduce overall uncertainty when executing a trajectory in the real world.

IV. Discussion

The demonstrated integration of a sampling-based motion planner with an accurate robot simulator enables path planning for tensegrity robots. This allows for paths that take into account both terrain effects and any other environmental obstacles, due to considering the dynamic

Fig. 7. A single set of particles that represent one belief distribution. This distribution is multi-modal The dynamics of the system naturally create these situations.

effects that contacts introduce on the robot. There are several pending research directions however to make the integration more effective.

A. Implementation Efficiency Concerns

One of the most obvious drawbacks discovered when planning with the physics engine is that the computational cost of planning is large. Especially in the case of planning under uncertainty, there is a lot of work to be done to make planning faster. This work takes advantage of parallelism to achieve faster times, but alternatives should be explored. It might be possible to find a different representation for the probability distributions that isn't particle-based. If this is possible, much of the computational cost can be reduced. Another possible direction is looking into more approximate models of tensegrity robots for long horizon planning. Then, the full simulator can be used more as a verification tool than planning primitive.

B. Algorithmic Additions

Much of the integration between the simulator and the planner assumes that there is no knowledge of the other component. The planner considers the simulator as a "black box" that given a start state, an end state is provided as output. If more knowledge about the underlying workings of the simulator is given to planning, more efficiency may be gained. By maximizing the usefulness of each iteration of the planner, the resulting paths will have better quality. This additions could be further parallelization, biasing the search region using heuristics, or even moving into a replanning framework where the planning horizon is shorter, but planning is done in more frequent stages.

Another way to improve the integration is to better focus the search process to promising controls and integrating this high-level planning method with efficient local gaits that have been recently developed [3]. This work uses random control inputs to the robot, while more intelligent control inputs will more effectively move the robot. The question then becomes, what is the set of diverse local gaits that allows for locomotion in the largest amount of cases? This question is the focus of ongoing work.

REFERENCES

[1] K. Caluwaerts, J. Despraz, A. Işçen, A. P. Sabelhaus, J. Bruce, B. Schrauwen, and V. SunSpiral, "Design and control of compliant tensegrity robots through simulation and hardware validation," *Journal of The Royal Society Interface*, vol. 11, no. 98, 2014.

[2] R. E. Skelton and M. C. De Oliveira, *Tensegrity Systems*, 2009th ed. Springer, June 2009.

[3] A. P. Sabelhaus, J. Bruce, K. Caluwaerts, P. Manovi, R. F. Firoozi, S. Dobi, A. M. Agogino, and V. SunSpiral, "System design and locomotion of superball, an untethered tensegrity robot," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2867–2873.

[4] B. Mirletz, P. Bhandal, R. Adams, A. Agogino, R. Quinn, and V. SunSpiral, "Goal-directed cpg-based control for tensegrity spines with many degrees of freedom traversing irregular terrain," *Soft Robotics*, vol. 2, no. 4, pp. 165–176, 2015.

[5] M. Likhachev and D. Ferguson, "Planning Long Dynamically-Feasible Maneuvers for Autonomous Vehicles," *International Journal of Robotics Research*, 2009.

[6] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *IEEE TRA*, vol. 12, no. 4, pp. 566–580, 1996.

[7] S. M. LaValle and J. J. Kuffner, "Randomized Kinodynamic Planning," *International Journal of Robotics Research (IJRR)*, vol. 20, no. 5, pp. 378–400, May 2001.

[8] S. Karaman and E. Frazzoli, "Sampling-baed Algorithms for Optimal Motion Planning," *International Journal of Robotics Research (IJRR)*, vol. 30, no. 7, pp. 846–894, 2011.

[9] Y. Li, Z. Littlefield, and K. E. Bekris, "Asymptotically Optimal Sampling-based Kinodynamic Planning," *International Journal of Robotics Research*, 2016.

[10] Z. Littlefield, D. Klimenko, H. Kurniawati, and K. E. Bekris, "The Importance of a Suitable Distance in Belief-space Planning," International Symposium on Robotic Research (ISRR), Sestri Levante, Italy, Tech. Rep., 2015.

[11] NASA, "NASA Tensegrity Robotics Toolkit (NTRT)," http://ti.arc.nasa.gov/tech/asr/intelligent-robotics/tensegrity/ntrt.

[12] J. M. Porta and S. Hernandez-Juan, "Path Planning for Active Tensegrity Structures," *International Journal of Solids and Structures*, 2015.

[13] Y. Li, Z. Littlefield, and K. E. Bekris, "Sparse Methods For Efficient Asymptotically Optimal Kinodynamic Planning," in *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, Istanbul, Turkey, 2015.