Motion and Control System of the TETwalker robot

Alberto Ruiz Aguilar

***Abstract*— The TETwalker represents a revolutionary idea in robotics and structural architecture. It is a creative application of Addressable Reconfigurable Technology (ART), developed by NASA researchers at Goddard Space Flight Center working jointly with Langley Research Center. This highly integrated three-dimensional mesh of actuators and structural elements has the potential to autonomously change form to optimize its function, reconfigure into specific tools, and perform tasks in a wide range of terrain and environment.**

The purpose of the Tetrahedral Walker (Tetwalker) project is to extend current space exploration into regions currently inaccessible by traditional wheeled or humanoid robots. The Tetwalker consists of arms, or struts, that join at nodes to form a tetrahedron. The struts will be able to lengthen and shorten, thereby giving the Tetrahedron shapeshifting abilities. The robot will move by changing the strut length to change its center of mass, causing it to tumble.

In the next sections it can be described the motion and the control system as they are describing in the modern literature. The dynamics of such structures can be utilized for locomotion and that the best control system is based on the decentralized adaptive control.

I. INTRODUCTION

The general definition of a tensegrity structure is a structure that maintains a stable volume in space through the use of discontinuous compressive elements (struts) connected to a continuous network of tensile elements (cables), explaining the acronym tensegrity = tension + integrity. [4]

New paradigm in the mechanical design of locomotor robots, based on the concept of tensegrity. Tensegrity structures are volumetric mechanical structures composed of a set of separate rigid elements connected by a continuous network of tensional elements. Due to an intricate balance between the tensile and compression forces in the structure, the structure is maintained at equilibrium. When a moderate deforming force is applied at one point of the structure, only a transient change is effected in the global form, after which the structure once again returns to its equilibrium configuration. [6]

The TET-robot is different from conventional robots because it does not use legs or wheels for locomotion. Rather, it moves by changing its center of mass, which causes it to tumble in a desired direction. Each TET-robot consists of struts that are connected at nodes. The struts are capable of extending and retracting, which changes the TET-robot's center of mass causing it to tumble.

Lightweight telescoping struts are attached at each end to pivoting nodes to allow movement over a wide range of angles. Motors within the nodes control the telescoping struts, allowing specific sections of the tetrahedron to lengthen or shorten, changing its center of mass. This enables the tetrahedron to maneuver in a controlled flip-flop motion by toppling over in alternating directions. The tetrahedron (tet) has four nodes and six expandable struts. It walks by extending certain struts, changing its center of gravity and falling in the desired direction. Currently, the basic structure, the tetrahedron, is being modeled as a cooperating/collaborating 4-agent system with an agent located on each node of the tet. (An agent, in this context, is an intelligent autonomous process capable of deliberative and reactive behaviors as well as social and introspective behaviors.)

II. MOTION

Centers on designing a constrained multi-agent system that efficiently and effectively allows a simple tetrahedral rover (a rover consisting of a single tetrahedron) to move autonomously on a surface and maneuver past obstacles it may encounter (such as a rock, wall, incline, cliff, or ceiling) in order to achieve a specified goal. Each simple tetrahedral rover will be autonomously controlled by a group of four agents acting in a coordinating and collaborative fashion.

Three methods of actuation are possible in a tensegrity structure: strut-collocated, cable-collocated, and noncollocated actuation. In strut-collocated actuation, the actuators are responsible for altering the strut lengths. In cable-collocated actuation, the structure is modified by changing the effective rest length of the cables. In noncollocated actuation, actuation is applied between two struts, two cables, or a strut and a cable. [3]

For this phasing the simple tetrahedral walker was considered through four increasingly complex systems: the node, the inchworm, the triangle and the tetrahedron. There are two types of emergence: emergence of autonomous behaviors in a simple tetrahedral structure arising from the progressive composition of autonomous behaviors realized by going systematically from a dot to an inchworm to a triangle to the simple tetrahedron. This is the major type of emergence hoped for in our current research. The emergent autonomous behaviors of more complex tetrahedral structures, realized from the composition of the individual autonomous behaviors of the basic building blocks (the tetrahedra), is the second type of emergence.

In order to better understand the problems of reactive motion and node adaptability in tumbling motion, a triangular configuration has been considered. Three agents located on the corners control the triangles motion by changing the lengths of the struts connected to the corner on which they are located. The triangle moves in a linear manner, reaching a target along the line defined by an xcoordinate. The triangle moves by extending struts so that its center of mass passed the front base node (see Figure 1,2,3,4).

An important goal for the triangle was not only for the system to successfully move, but also for it to do so in the most efficient way possible. In this scenario, efficiency is measured by absolute value of change in length in all struts during motion. The most efficient motion involved maximum extension of the far-top strut, minimum but positive extension of the top-close strut, and negative extension of the base strut.

The primary emergent design of these systems is the Tetrahedron. In this case four agents control six struts to move the structure to the goal location. [5]

Initially the tetrahedron was considered without obstacles or inclines (roving on a flat surface towards an (x,z) goal). In this case each agent followed the following logic: First, the node checks to see if it is part of the base (if the yvalue of its location is equal to zero). If this condition is satisfied, the agent goes on to see which base node it is. If its location is farthest from the goal, then it tells the strut connecting itself to the top node, to extend to a set length (between two and three times the minimum strut extension). Also, it tells the two other struts it is connected to, to contract to the minimum length for a strut. If two nodes are both equidistant and farthest from the goal, one is selected at random to extend to this length, the other acts like the following nodes. If a node satisfies the first condition, but not the second, it informs the strut that connects itself to the top node to extend to a different set value (just a little more than the minimum). The extension is done in a step-based manner and is halted when the tet falls over (the top node is constantly checking to see if the tet has fallen over, or, reached the goal). Positions are reevaluated upon falling, motion is stopped when the top node has reached the goal. This motion succeeded in reaching the goal.

Many algorithms were added into the tet AI(Artificial Inteligenci) to incorporate obstacle avoidance methods into the tetrahedron. In the old situation the top node had no job. The top agent is now the path planner for the tet. The reasoning of each agent has now become the following: The two base, not farthest nodes based on the current subgoal (initially equal to the goal location) do exactly what they used to in the old AI. The other base node extends the strut between itself and the top node to a length that is not enough to make the tet fall, but long enough to be very ready to fall when the new subgoal is determined. When the new subgoal is determined (which is usually in the same direction as the old subgoal) the positions of the base nodes are reevaluated and the old movement sequence is initiated by them.

Meanwhile, the top node decides on the direction in which the tetrahedron should be moving. Along with a few variables inserted to determine completion of certain tasks or successes, there is the new stored knowledge in each agent of the subgoals location, and two different maps of the environment (1000 x 1000 two dimensional byte arrays). Both maps are initialized by the agents to all 0s, or unknown, when the agents are created. The first thing a node does when it becomes a top node is to communicate to all the other nodes to not move into a falling configuration yet, until I decide where we are going. Then, the agent requests a two dimensional array (map) describing the area within its vision [right now a square with a side length about three times the length of a contracted strut, but, the AI would still work equally well (with very minor tweaks) with any shape vision and with shadows or other limits to vision]. That small map is then integrated into the agents large map it has been keeping. This history is kept in exactly the same way as the inchworms is. This process is repeated for two maps, one that is a realistic interpretation of the environment, and one in which the obstacles are enlarged. The new maps that the agent has compiled are then sent to the other agents to be used as their maps when they become the top agent.

Once the cartography is done, the agent looks around in a square a little bigger than the area of vision for the edge point, or 3, that has the smallest sum distance to goal and the center of the base of the tet (the point closest to the line connecting the tet and the goal). For this consideration the tet uses the map with enlarged obstacles as to pick a point that is likely to be safe to get to. The agent next checks to see if the path to that point is clear by checking not only the line to that point (as the inchworm does), but a series of short lines perpendicular to that line (in the case of this simulation, they are only somewhat perpendicular) for obstacles (on the real map). If there is a clear shot to the point, that point is set as the subgoal and the other agents are told that they can make the tet fall over. If the direct path to that point is not clear a more complex consideration of the possibility is executed. [1]

First, the agent identifies the corners of the square obstacles in a window of the enlarged obstacles map slightly larger than the square it has just seen. Next it uses the same logic as the inchworm (but with the previously explained new method to check direct paths) to see what the best path to the point is. If a path is found, the first point on the path is set as the subgoal and the other agents are told to make the tet fall over. If instead, the agent fails to find a path to the subgoal, all the threes in the area close to that subgoal are relabeled ones, or free space, and a new subgoal is determined. If all of the threes are eliminated, the threes are regenerated and the process is repeated without ignoring the threes that are close to eliminated threes. If that is not sufficient, the entire map is considered. Through these processes a subgoal is eventually established and a new node reaches the top, and begins planning.

Throughout all of this the nodes are checking to see if the goal has been reached, and, if it has been, communicating this to the other nodes stopping the system, and declaring the mission a success.

III. CONTROL SYSTEM

Therefore, it was decided that a control system based on the Decentralized Adaptive Control (DAC) algorithm developed by Homayoun Seraji should be implemented, after it was proven to be applicable to the TET-robot. [2]

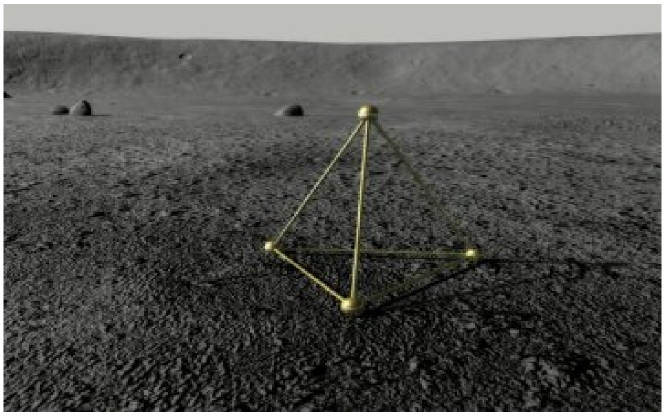

Fig. 1. Frames extracted from video of Tetwalker(1)

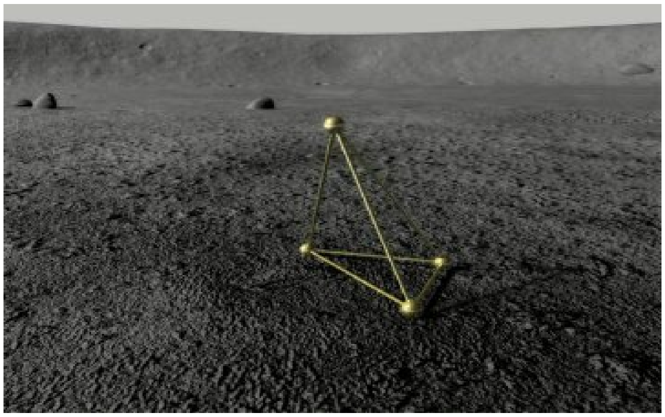

Fig. 2. Frames extracted from video of Tetwalker(2)

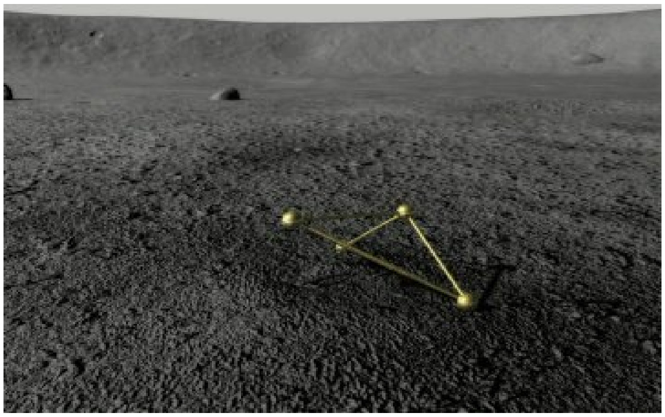

Fig. 3. Frames extracted from video of Tetwalker(3)

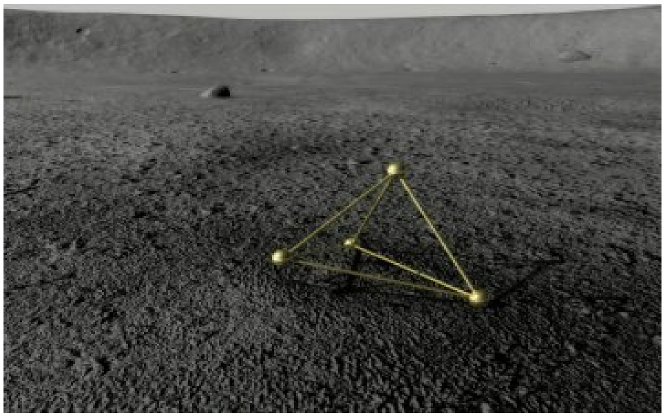

Fig. 4. Frames extracted from video of Tetwalker(4)

The major benefits of the DAC system are best stated by Seraji as that the ”knowledge of the manipulator dynamic model and parameter values or the payload parameters are not required” and that this ”scheme is computationally fast and is amenable to parallel processing implementation within a distributed computing architecture, with one microprocessor dedicated to each joint.” These two reasons embody why this control scheme is ideal for the TET-robot, as it would be very difficult to derive a set of equations to completely describe the robot. Also, the DAC system is ideal for TET robot because it would allow each strut to act somewhat independently from each other. This control scheme also fits perfectly with the future goals of adding more struts to the TET-robot. Since the DAC system would be implemented on the microcontroller of each strut, the addition of more struts would only require modification of the equations describing the gait to be performed by the TET-robot, rather than completely redefining the control system. It is expected that by implementing the DAC system the centralized system will only be responsible for informing the struts which gait to assume.

IV. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In traditional robot design, the goal of control is to ensure that each joint is precisely controlled and tracks a desired joint trajectory, although the physical structure of a robot usually includes dynamic interactions between the motions of multiple joints. Thus, the control often seeks to decouple dynamic interactions between the individual joints. While such designs often lead to successful control strategies, from the perspective of the mechanical structure, they are often not fault-tolerant. Thus, for example, in a quadruped robot, a broken knee joint may drastically impair the ability to produce gait. In a tensegrity robot, actuation at one location of the structure produces motion at multiple locations. The dynamics are even more coupled than in a traditional robot. This feature gives the structure a high degree of fault tolerance. If an actuator is damaged, another may be used to make up for its function.

The fact that application of a force on one part of the structure causes a global deformation in the structure also presents other benefits. One actuator can be used to actuate multiple cables, which leads to the possibility for a small number of actuators to cause a global movement pattern and for multiple subsets of actuators to be used to produce the same behavioral outcome.

Tensegrity robots also have other advantages. They can be lightweight, due to the fact that the structure achieves its rigidity based on a high number of tensile elements and a relatively small number of rigid elements. Moreover, as only a small number of actuators are used relative to the number of degrees of freedom, this can lead to additional reduction in weight. Tensegrity robots also have a high strength-to-weight ratio and are effective at absorbing shocks.

In addition, tensegrity robots have the possibility for low-volume stowage, self-deployability, and reconfigurability. These are new features in the realm of robotics, which have not been easily achievable using conventional technology. While the utility of these features may be limited to certain application domains, they nonetheless broaden the range of possibilities for robots.

While the TETwalker is currently controlled remotely, also under development is a synthetic neural system to enable the TETwalker to function autonomously. This neural system will allow the TETwalker to adapt and actively reconfigure itself according to its environment and recognized needs. Like the physical architecture, the neural system has a three dimensional node-driven architecture.

B. Future Works

Future developments will reduce size using Micro-Electro-Mechanical Systems (MEMS) and then further using Nano-Electro-Mechanical Systems (NEMS). With this refinement, even greater control and agility will be possible.

Beyond that, the techniques used in these systems will be applied to much more complex structures. These architectures will present many new challenges for the multi agent systems. Unlike in the tetrahedron, at any point in time no one agent will be able to know everything that is going on, and must therefore focus on its neighbors. A large number of agents properly implementing this strategy will not be able to control the system as they cannot see the big picture for the structure. Therefore a hierarchy must form in order to control goals as well as implement reactive algorithms. These systems have given us some guidance in terms of how to approach these problems, but the challenges that will arise will be far more complex.

V. ACKNOWLEDGMENTS

The author gratefully acknowledge the contribution of ”Robótica Autónoma Móvil”'.

REFERENCES

[1] C. Sebens, ”‘An agent-based tetrahedral walker”', *Aerospace Conference IEEE*, pp. 7, 2006.

[2] B. Solomon, P. D. Olivier, ”‘Implementation of a Decentralized Adaptive Control System for the 12-TET Walker”', Thirty-Ninth Southeastern Symposium on System Theory, pp. 102-103, March 2007.

[3] C. Paul, F.J. Valero-Cuevas, H. Lipson, ”‘Design and control of tensegrity robots for locomotion”', IEEE Transactions on Robotics and Automation, Vol.22, Iss.5, pp. 944- 957, Oct. 2006.

[4] C. Sultan, ”‘Tensegrity Structures Research Evolution”', 45th IEEE Conference on Decision and Control, pp. 2294-2299, 13-15 Dec. 2006.

[5] M.C. de Oliveira, R.E. Skelton, Chan Waileung, ”‘Minimum Mass Design of Tensegrity Towers and Plates”', 45th IEEE Conference on Decision and Control, pp. 2314-2319, 13-15 Dec. 2006.

[6] R.E. Skelton, R. Adhikari, J. P. Pinaud, Chan Waileung, J.W. Helton ”‘An introduction to the mechanics of tensegrity structures”', Proceedings of the 40th IEEE Conference on Decision and Control, Vol.5, pp. 4254-4259, 2001.