Extensions to the NASA Tensegrity Robotics Toolkit

CISC499 Projects Supervised by Dorothea Blostein
Winter 2015

Tensegrity (tensional integrity) is a structural principle popularized by Buckminster Fuller: a network of elements under compression is held in place by elements under tension. Tensegrity structures are both strong and flexible due to the dynamic interplay between tension and compression forces. Applications of tensegrity include

art	www.youtube.com/watch?v=5ALMEq219Nc www.youtube.com/watch?v=xDNohDRWTvU
architecture	www.youtube.com/watch?v=IeWWPAckC5U
biological modeling	www.youtube.com/watch?v=uMug6XzP1R4 www.youtube.com/watch?v=AT5fsO95-qg
robotics	www.youtube.com/watch?v=wR0AlIwEgSE

The NASA Tensegrity Robotics Toolkit (NTRT), available since June 2014, is a public domain collection of C++ and MATLAB software modules for the modeling, simulation, and control of Tensegrity Robots. ti.arc.nasa.gov/tech/asr/intelligent-robotics/tensegrity/ntrt

The goal of this CISC499 project is to extend NTRT in one or more the following ways:

- Refactor NTRT to be more robust and easier to maintain. Tasks include: reduce repetitions (clones) in the build scripts and code; use declarative rather than procedurally coded specification of the connections in a tensegrity structure; improve support for unit testing and dependency injection.
- Improve documentation, both for novice and expert users of NTRT.
- Take advantage of multi-core processors.
- Create a suite of unit tests for the core NTRT functionality.
- Add the ability to simulate adaptation in tensegrity structures. For example, an adaptation rule can specify that a tensioned element becomes stiffer in response to being under a lot of tension. Adaptation produces structural learning in a tensegrity network, with interesting analogies to learning in a neural network. Two of Professor Blostein's recent MSc students, Adrian Muresan and Slava Jdanov, coded simulations of adaptive tensegrity structures. They wrote stand-alone code because NTRT was not available at the time they did their work.

One or more students can undertake CISC499 projects on this topic. Students must be able to work independently, with the confidence to work on a large software project. While the NTRT code base itself is relatively small (around 8000 lines of code), it relies on enormous libraries such as the Bullet physics engine. Students need to have (or quickly acquire) experience with OS X or Linux.