MIT CSAIL ALLIANCES

Drake: Model-Based Design and Verification for Robotics

Case Study

Equipped with accurate sensors, deep learning models and algorithms, soft skin and fingers, and much more, robots are becoming increasingly capable of performing tasks autonomously. But while robots are performing more and more complex tasks that could help people at home and at work, these experiments are typically limited to individual robots in a lab setting, and it is challenging to apply the same engineering that makes these robots perform at scale in the real world. What will it take for robot locomotion and manipulation to fully take flight?

Enter the dragon: specifically, Drake ("dragon" in Middle English), an open-source C++ / Python toolbox hatched by the Robot Locomotion Group at the MIT Computer Science and Artificial Intelligence Lab (CSAIL). Drake is a model-based design software that brings together a multibody dynamics engine, systems framework, and optimization framework, providing roboticists the tools they need within a single application. Since its inception, Drake has grown into the robust behemoth it is today with development support from the Toyota Research Institute, DARPA, the National Science Foundation, the Office of Naval Research, Amazon.com, and The MathWorks.

Drake: MIT CSAIL Origins

In CSAIL, Professor Russ Tedrake and his students in the Robot Locomotion Group started developing the main software infrastructure for Drake back in 2005. A few years later, Prof. Tedrake tried to share the software more broadly because of a real need he felt for it in teaching his MIT course about Underactuated Robotics.

"Every year, students would come up with fantastic ideas for their final projects, but then they would spend so much time getting a basic simulation working they rarely got very far with the more interesting ideas," he says. "Giving them code to get started really improved the quality of the projects!"

Then when the DARPA Robotics Challenge was held from 2012 to 2015, things started to get much more serious for Drake. The challenge of programming a nearly autonomous robot in a disaster response scenario opened Prof. Tedrake's eyes to the advantages of software engineering, and also the limitations programming a single robot imposes.

"I had a stack of build servers in my office," says Prof. Tedrake, adding that they were a real pain to maintain. "Back then Drake was still written in MATLAB, with just a little C++ in performance critical components. This was great for prototyping, but hard for achieving professional-grade code…so we limped along with something I wrote myself."

Around the same time, the Toyota Research Institute was forming into the entity it is today.

"When TRI was born, I saw this as a chance to finally get to mature code," Prof. Tedrake explains. "It's been amazing to learn from professional software engineers and watch the migration from research prototype to professional code."

(continued)

Software Development with the Toyota Research Institute
Prof. Tedrake joined TRI as one of the first employees, working with Dr. Gill Pratt, who started TRI. "From the very beginning, Gill and I were both very excited about the impact that high-quality simulation could have on the field."

The industrial research lab and its strong relationship with CSAIL led to Drake's core development as a toolkit for analyzing the dynamics of robots and building control systems for them, so that their designs and analyses are optimized. Even very complex dynamics of robots can be simulated in a way that shows the underlying structure and equations, rather than the black box common in most simulations.

TRI, which recently announced its continuing AI research collaboration with MIT, has invested heavily in the software and hardware tools that keep the vision for Drake alive and provide mature tools that are all open-source.

"TRI has a very strong relationship with CSAIL," Prof. Tedrake explains. "TRI provides significant funding, and has very productive collaborations with faculty. Setting up TRI to be just down the street from MIT (and Stanford, and University of Michigan) was an extremely good, strategic decision."

The Benefits of Open-Source Technology
With the three main core components of multibody dynamics, systems framework, and optimization framework, Drake pushes research in advanced algorithms for robotics to the cutting edge, allowing for further perception, planning, and control. Drake provides an interface to Python

TRI's commitment to sharing this open-source software with the world will continue to help roboticists program and test their prototypes of individual robots and whole fleets of robots at scale.

"I believe that open-source is ultimately the best way to have a broad impact in the community, and more users/developers ultimately means more contributions and more powerful tools," says Prof. Tedrake. "Quite a few of the Drake developers chose TRI to work in open-source instead of other great companies, where they would have to keep their work more hidden."

Now that Drake is ready for industrial strength robotics research and engineering applications, Prof. Tedrake is looking forward to sharing the toolbox and welcomes your own open-source contributions and feedback.

"The problems in robotics are hard enough," he says. "For the parts that we know how to implement, we should work together as a community to get one or more gold standard implementations, and give ourselves the solid foundation that we need to break through the next research barriers."

Read more about Drake and TRI in this blog post by Prof. Tedrake.