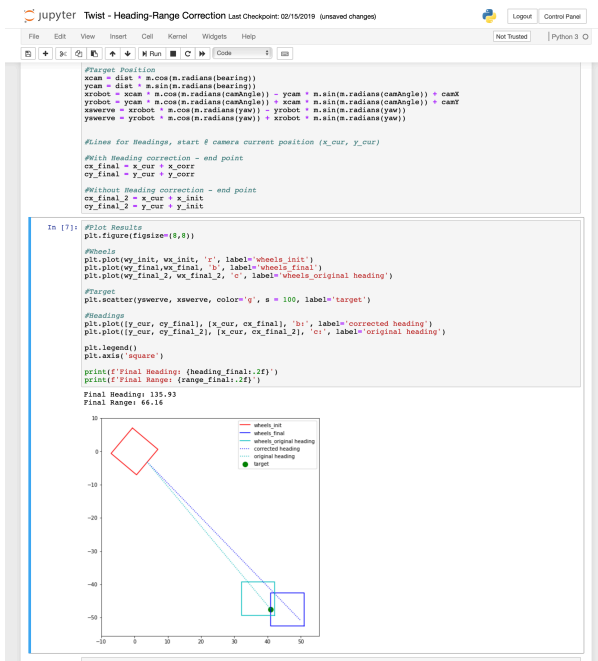


ease of logging in and being able to quickly test any idea anywhere.

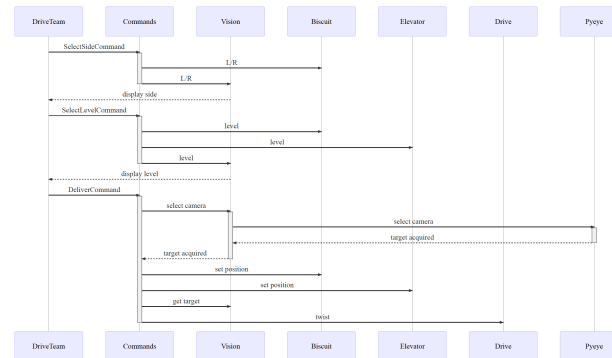


We are pleased to make TCT and the Grapher available as open source at <https://github.com/strykeforce>. More resources are available at <https://www.strykeforce.org/resources>.

Software Engineering

As members of Stryke Force, we learn how to work together using real-world tools and processes that will benefit us during college and our future careers.

This year's *FIRST®*DESTINATION: DEEP SPACE code is written in Java, an **object-oriented programming** language and leverages WPILib command-based programming. Our use of **system sequence diagrams** speeds up development and testing since all events are carefully planned in advance.



The *FIRST®*DESTINATION: DEEP SPACE and Pyeye projects use GradleRIO, the open-source build automation software. Stryke Force programmers use Gradle to build projects, deploy code to the RoboRIO, and manage dependencies.

Stryke Force uses GitHub and Git for version control. The programming team adopted the **OneFlow Git Branching Model and Workflow**. The programmers maintain one master branch and create pull requests from feature branches. For example, when working on autonomous paths, a programmer would create a new branch in which they would add and modify the autonomous code. The programmer would then submit a pull request to the master branch in the Stryke Force repository. Pull requests undergo careful peer and mentor review before being merged into the master branch.

Team 2767
Stryke Force
 Est. 2009
 Kalamazoo, MI



Software Development Team
2019

*FIRST®*DESTINATION: DEEP SPACE Presented By The Boeing Company

www.strykeforce.org

www.github.com/strykeforce

Software Designed for Drivers

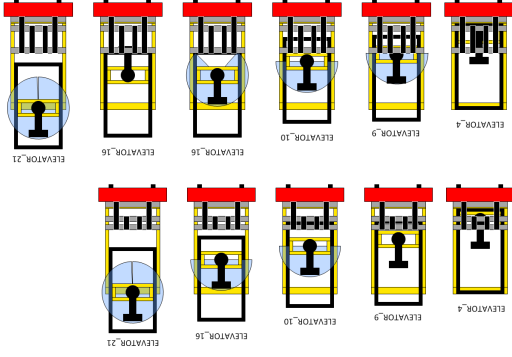
Stryke Force is proud to present the control system software for **Air Tight**, our robot competing in *FIRST®*DESTINATION: DEEP SPACE Presented By The Boeing Company.

We strive to meld high-performance hardware with custom software to provide our drive team with the best robot possible. Our **Third Coast Sverve Drive** software has historically provided unmatched maneuverability and response.

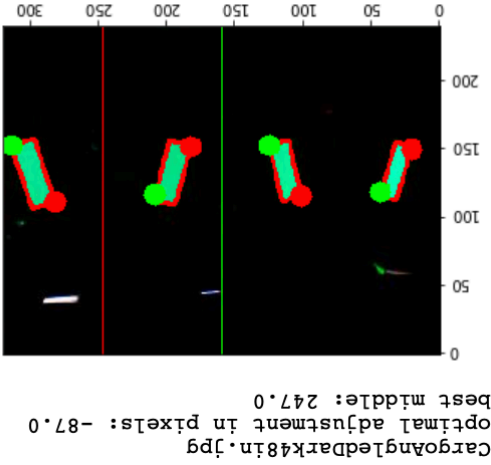
Precision Control Systems

Stryke Force delved deep into motion profiling this year. Using the **CTR-Electronics Talon SRX Motion Magic** functionality, the programming team controls the motion-profile using acceleration, velocity, current limit, and PID parameters. This feature in the software allows the robot to make precise and repeatable movements in all subsystems.

“Robot Safety Classes”



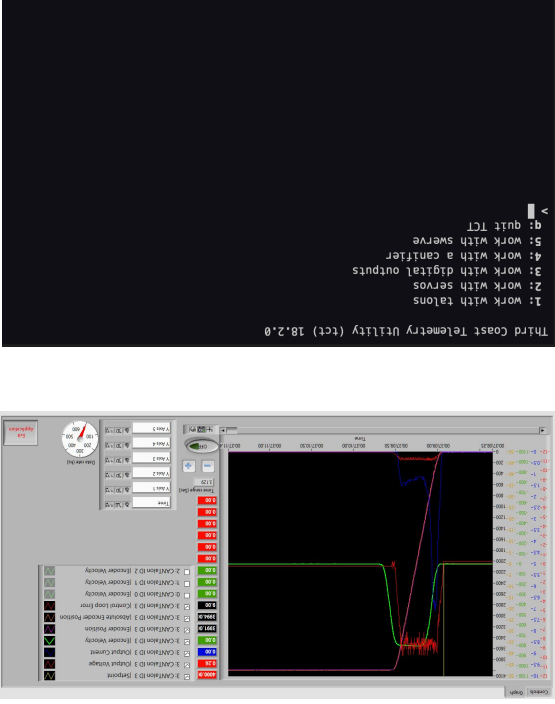
At Stryke Force, safety is our primary concern so it's only natural that safety rules apply to all, human and other-



“Twisting our way through the sandstorm”

wise: robot. Programmers are using a custom **Safety Sub-system** that interfaces with the Elevator, Biscuit (arm rotation), and Intake subsystems. Every 20ms, all three subsystems update their current state and then the Safety Subsystem runs through a matrix of legal and safe positions updating Talon SRX forward and reverse soft limits. Since the axes cannot pass the soft limits, the programmers ensure the robot's safety.

The software team focused on helping our driver perform during Sandstorm. In addition to pathfinding using Pathfinder, we're implementing **computer vision** (CV) to guide our driver through Sandstorm. Since vision processing significantly affects CPU performance, we decided to use a **Raspberry Pi co-processor**. Our custom Python CV application *Pyeye* uses OpenCV to calculate the range and bearing to targets which we then publish to NetworkTables. After transforming the range and bearing from the camera coordinate system, the robot autonomously approaches the selected target and places the Game Piece in a movement called the “**twist**”. Borrowed from the Robot Operating System (ROS), the term describes the motion of a yaw transposed onto a linear movement.



Custom Development Tools

The software team also builds and maintains custom development tools. These tools are used by the entire team to test and tune drive systems, actuators and sensors. The Stryke Force **Grapher** application allows the team to log and analyze the robot performance and perfect the tuning by plotting data received from the RoboRIO. Stryke Force is able to chart almost any data possible from the RoboRIO. **Third Coast Telemetry** (TCT) and the Grapher applications provide Stryke Force with deep insight into robot performance and are invaluable to the development process.

Similar to the Grapher, **Keeper** logs data recorded from the robot during the execution of a command. The data is then made available through a Dash app using PostgreSQL and Plotly. Stryke Force programmers are using a **Jupyter Notebook server** this year to test CV applications and algorithms, coordinate transformation math, and any other ideas that come to mind. All of our programmers have enjoyed the