

Satellite Attitude Testbed Software Automation (SATSA)

Ava Crocker, Maxwell Caiazza

Faculty Advisor: Dr. Siddhartha Bhattacharyya, Dept. of EECS, Florida Institute of Technology

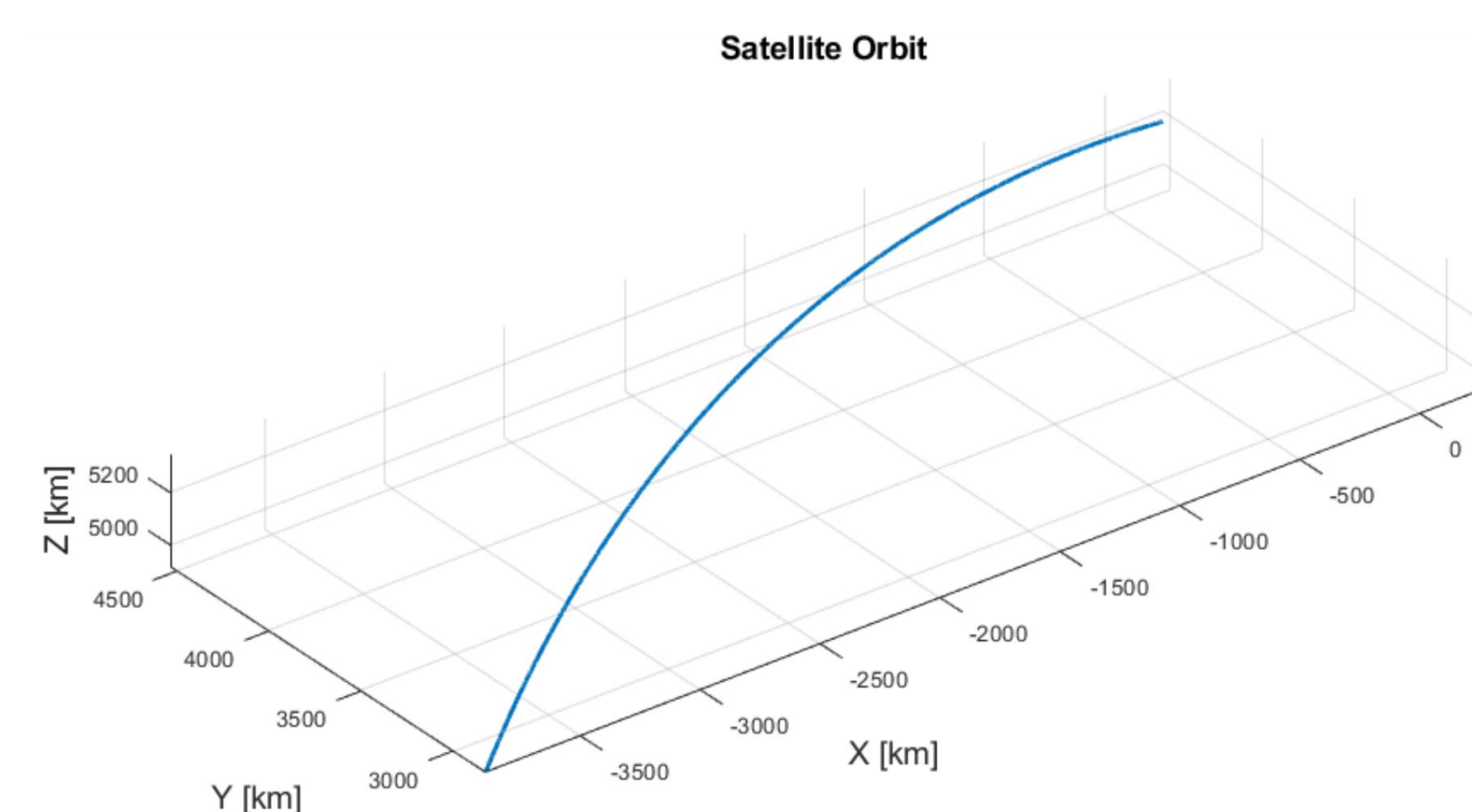
Client: Dr. Camilo A. Riano-Rios, Dept. of Aerospace, Physics, and Space Sciences

**ENGINEERING & SCIENCE
STUDENT DESIGN SHOWCASE**

FLORIDA TECH

Goal

- SATSA's mission is to elevate the software usability of a satellite simulator focused on an Earth-orbiting satellite's guidance, navigation, and control.
- To streamline the simulation process, a system will be developed to automate data input and node creation in ROS 2.
- Documentation and procedures will be outlined allowing contributing engineers without prevalent software backgrounds to easily add in necessary calculations and operations without technical knowledge.



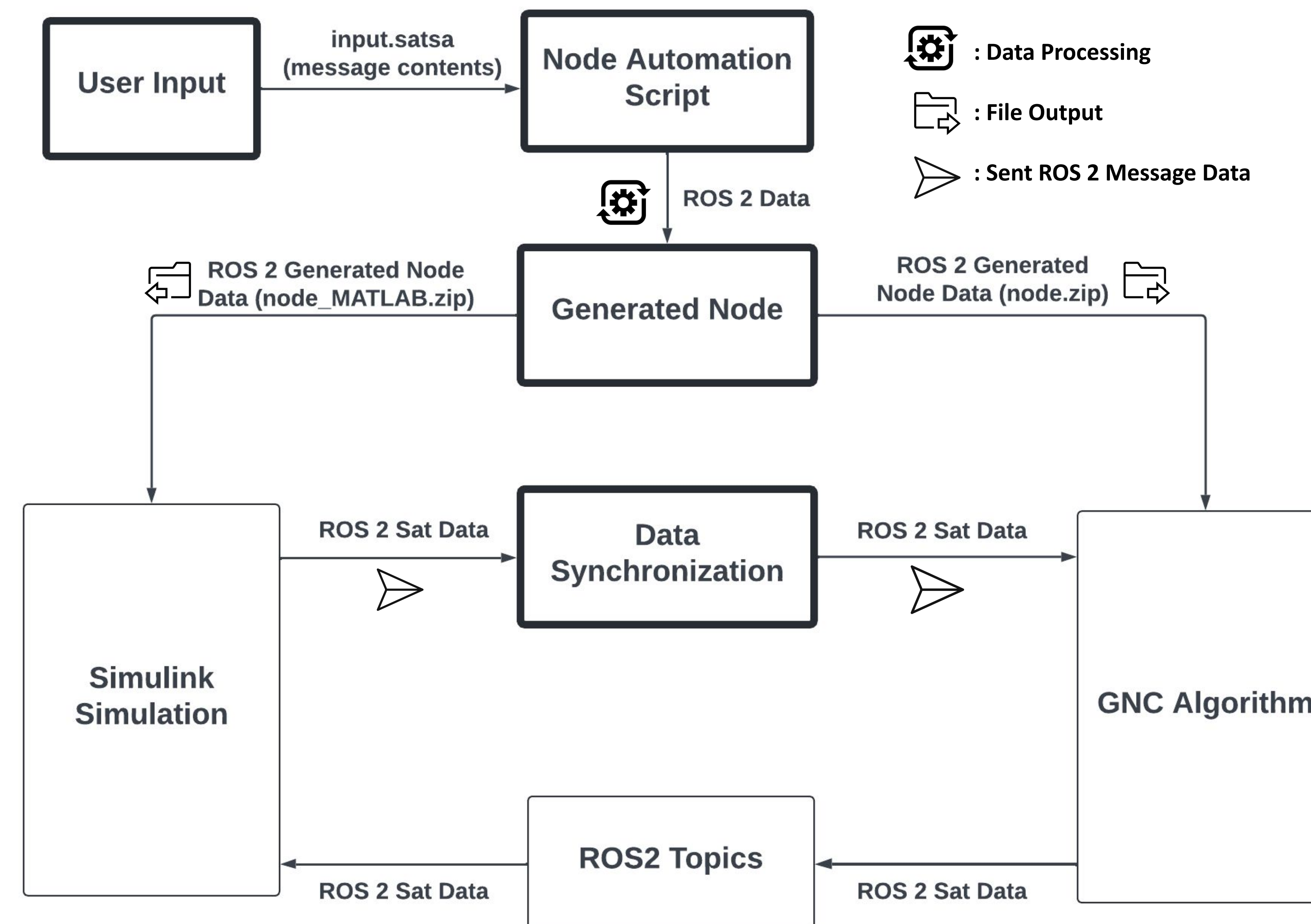
Methods

- Project development executed in agile structure
- Prioritize end-user and long term version functionality with ROS 2 and Python
- Abstract as much of the process as possible.
- Provide ample tools to the end user to solve problems without the software creators.

Results

- ROS 2 node successfully generated
- All requirements satisfied
- Satellite data ingested and sent through software-loop
- Engineering test team successfully followed documentation to generate data

Architecture

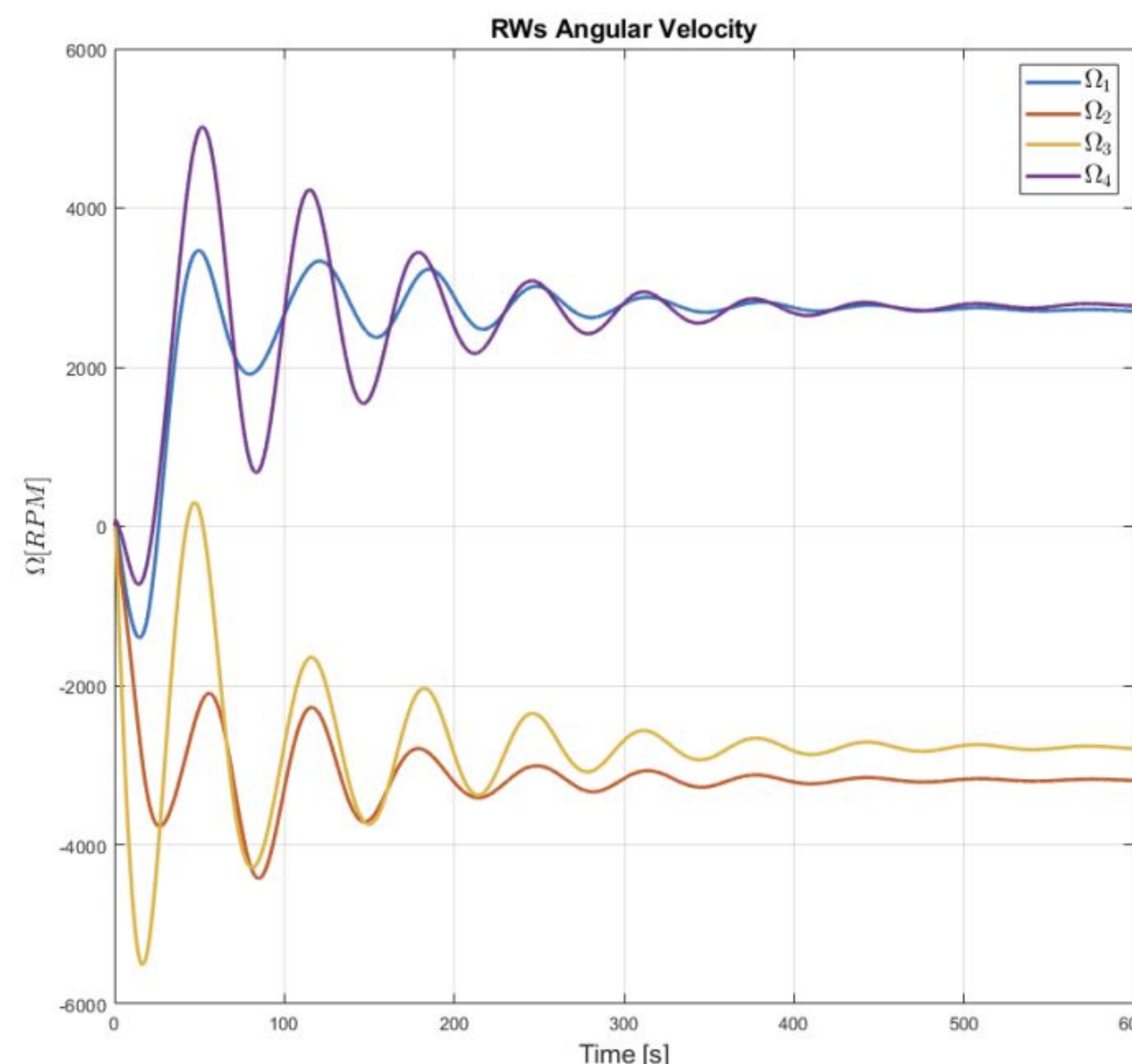


Tools

- **Python 8:** Python was used to develop all node generation automation. ROS 2 supports up to Python 8.
- **MATLAB:** MATLAB was used to initialize ROS 2, start up the simulation, process data, & host Simulink
- **Simulink:** Simulink was used to host the simulation and backend data transmission to generate plots
- **ROS 2 (Jazzy):** ROS 2 was used to create the satellite node, contain/parse specific data, and generate the desired output
- **GitHub:** GitHub was used for repository and version control



Generated Data



Software Engineering Insights

- Build system from the ground up
 - Processing and runtime could have been improved with specific data structures
 - Focus on modularity and project structure
 - Update data transfer to arrays

Future Work

- Incorporating further automation where the automated node is immediately sent to the software loop.
- GUI Implementation for taking user input while keeping .satsa file functionality in the background.
- Rework original program structure further to improve program time and efficiency