OPEN SOURCE SOFTWARE FOR SIMULATING COLLABORATIVE NETWORKS OF AUTONOMOUS ADAPTIVE SENSORS

Ryan Linnabary, Andrew O'Brien, Graeme E. Smith, Christopher Ball, and Joel T. Johnson

ElectroScience Laboratory
Department of Electrical and Computer Engineering
The Ohio State University, Columbus, OH

ABSTRACT

Collaborative networks of small satellites will form future Earth-observing systems. Maximizing the science value of measurements from such systems will require autonomous decision making with regard to management of limited resources (i.e. power, communications, and sensor configuration). The complexity of this decision space warrants the creation of software tools to aid users in efficient modeling and simulation of collaborative remote sensing networks. In this paper, we present a new open-source software library and tool–set that has been specifically designed for simulating such networks. Details of the object-oriented C++ library are presented with results from example simulations to confirm that it is able to address this challenge. The software tools developed offer enhanced simulation capabilities to developers of future observing system simulation experiments (OSSEs) with collaborative networks of adaptive sensor platforms.

Index Terms— Autonomy, Sensor Network, Software, Remote Sensing, OSSE

1. INTRODUCTION

The use of large, heterogenous networks of small satellites for remote sensing has been extensively reported as the future for Earth observing systems [1, 2]. A taxonomy of sensor network designs can be formed by categorizing them into classes according to overall complexity [3]. The simplest class of networks is formed of sensors that collect basic data independently. The second class is composed of sensors that coordinate their measurements in some way to aid in data fusion and enhanced value or decision making. The most advanced class encompasses highly autonomous systems in which there is collaborative system level reconfiguration of sensors in response to observations (i.e. adaptivity). It is this final class of collaborative sensor networks that may offer the greatest capabilities, but it is also the class currently at the lowest technology readiness level.

With the rapid increase in both the reliability and capability of small satellites, the formation of operational collaborative and autonomous networks in space is quickly becoming feasible. Concurrently, a need is emerging for the design of advanced OSSEs that incorporate collaborative small satellite networks. OSSEs provide critical information regarding long-term system behavior, optimizations, and alternative configurations [4]. Standard OSSE software supports platforms with fixed sensor parameters including observation geometery, spatial-temporal sampling, resolution, and observation metrics. However, intelligent and adaptive sensor platforms may vary all of these parameters in response to the scene observed or to queues from other sensor or model data. The complexity of collaborative, adaptive and resource constrained networks of remote sensor platforms poses a barrier to understanding the feasibility of such systems and navigating the cost/benefit design space. Therefore, new software tools are needed to simulate these systems to aid in their understanding.

The open-source C++ library Collaborate [5] is currently under development to provide support for the simulation of collaborative networks of adaptive sensor platforms in future OSSEs. Details of the software interface and functionality are introduced here, along with motivation for its inclusion in larger simulations. The library is designed to quantify the improvements (i.e. science value) of the usage of collaborative, autonomous decision making networks of adaptive sensors through rapid simulation and modeling. Collaborate simplifies observing-system simulation through constellation design involving diverse sensor platforms and visual analysis of network behavior. The library's object-oriented class hierarchy provides a straight-forward approach to incorporating user-defined algorithms into simulations. Results from example simulations are presented to illustrate these properties.

This research was sponsored by the NASA Advanced Information Systems Technology Program NNH16ZDA001N-AIST.

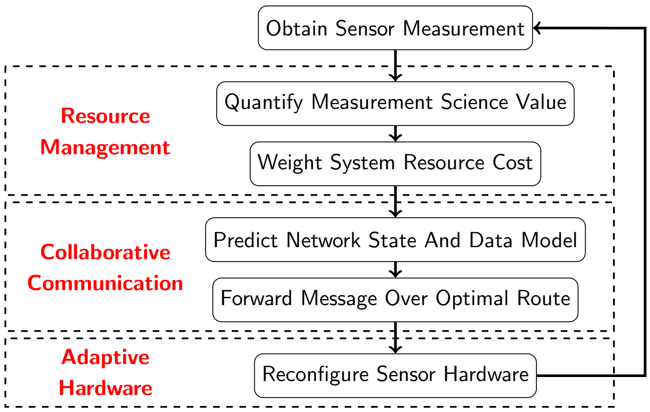


Fig. 1. Collaborative sensing model showing how autonomous adaptive sensor platforms communicate information to maximize science return from measurements.

2. COLLABORATIVE SENSING MODEL

The foundation of the software library design is the fundamental collaborative sensing model shown in Figure 1. Software components are organized to achieve the information flow shown in the figure. The three primary groups of components include resource management, collaborative communication, and adaptive hardware. Operations on a single platform, including resource management and hardware reconfiguration tasks, are generalized for multi-platform networks through Collaborate network communication algorithms.

In Figure 1, the model begins with a node in the network obtaining a sensor measurement. At this point, the satellite's actions are limited by system resource availability. Resource cost is especially critical for small satellites due to their limited power and computing resources. It is expected that the cost of subsequent measurements and communications is weighted against potential science value gain. Collaborate supports system resource optimization through native system model prediction and connection to user-defined algorithms.

Next, a platform deliberates collaboration with other members of the network through internal predictive modeling of satellite positions and geometry. These processes identify opportunities to improve measurement science value through collaborative communication. Communication efficiency is optimized by identifying channels which minimize the energy consumed in transmission within time limits dictated by the changing environment. This may involve maneuvering satellites to improve communication links or comparing time-varying communication parameters. Though the described sensing model may be replaced as network complexity grows, Collaborate communication algorithms will remain central to the design of autonomous networks in these simulations.

Finally, once the target satellite obtains the information, it adapts its behavior or its sensor hardware configuration to increase the science return of planned measurements. The nature of adjustments made in simulation depends on the hardware implementation, so Collaborate provides an object-oriented approach to sensor parameter reconfiguration. For example, the framework for Fully Adaptive Radar discussed in [6] has potential for extension to use on distributed sensors. Notions of cognitive hardware reconfiguration are closely aligned with the intended use of the library.

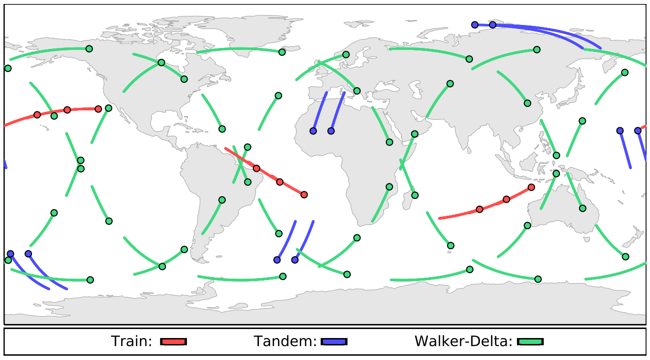
3. SOFTWARE OVERVIEW

The Collaborate Software Package provides (1) a C++ development library for OSSEs, (2) example C++ programs demonstrating observing-system construction, and (3) a Python package for analysis and visualization of outputs from the C++ programs. The development library and examples contain C++11 features and depend only on standard library components. The analysis tools are written in Python3 and depend on open-source libraries such as NetCDF4, Matplotlib, Cartopy, Numpy, and others. Developers can satisfy dependencies using the provided application container (Linux Docker image) or by following instructions in the README file. Collaborate is licensed under LGPLv3 and compiles as a dynamically-linked shared library for use on Linux systems. Its source code is freely available through a Git repository at [5]. The code conforms to Google's Style Guides and Python Enhancement Proposals 008 and 275. Documentation generated by Doxygen is published to a public Application Programming Interface.

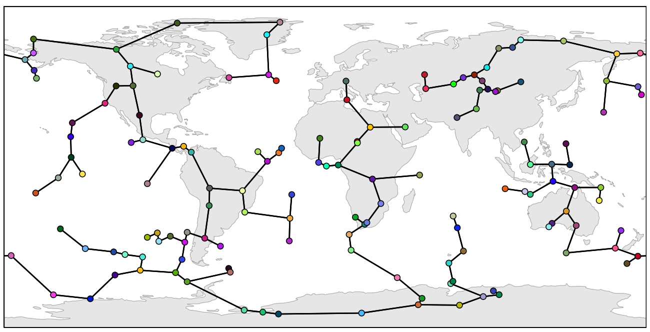
Functionally, the software package helps the user with the following tasks: rapid definition of satellite observing systems; introducing autonomous network algorithms for resource management, collaboration, sensor hardware adaptation; and the capture, analysis and visualization of the operation and results from the simulations.

Defining Satellite Observing Systems

The Collaborate development library aids in constellation design and configuring satellites with accessories, including RF interfaces, power subsystems, and logic processing units. Standard orbital models are used to define satellite motion. Constellations are created by distributing modified duplicates of Two-Line Element (TLE) Sets in patterns defined by the user. Figure 2(a)) illustrates how the library expedites the design of complex constellations, such as trains (red), tandem formations (blue), and larger distributions in several orbit planes (green).



(a) Example constellations generated using the library.



(b) Minimum spanning tree - the unique minimal sum of line-of-sight distances between satellites.

Fig. 2. Examples of satellite sensor network construction and programming.

Satellites are equipped with interfaces for sensing and communication. Sensor interfaces link satellites to the underlying data sets for measurement, based on antenna/sensor patterns and the intersection of its pointing direction with the Earth's surface or atmosphere. Communication takes place between two satellites with related interfaces. The library provides standard communication hardware models for UHF, free-space optical, and commercial satcom. Data buffers are implemented as byte arrays, but communications occur at the packet level. The most important communication parameters identified for simulations include the link bandwidth, latency, error rate, tx/rx power, doppler tolerance, path loss, and data rate. These parameters (and others) inform decisions made by the communication algorithms, allowing the simulation of realistic inter-satellite communication links.

Autonomous Sensor Network Behavior

COLLABORATE provides many communication algorithms to facilitate the autonomous behavior described in Section 2. They offer developers a head-start moving the information required by other optimization algorithms. Additional algorithms may be defined internally or as external programs. Though the library implements basic hardware for sensors and power subsystems, it has been designed for seamless customization with user-provided optimization tools.

Collaborative networking is achieved using the library's local data structures and algorithms. Examples include trees and graphs for modeling network connections based on geometry, motion, communication parameters, system health, and resource availability. Several well known graph and tree algorithms are provided, including Dijkstra's shortest path and Prim's minimum spanning tree (Figure 2(b)).

An advanced predictive algorithm is developed to accommodate the dynamic nature of space networks. Its practical demonstration is provided later in Section 4. The algorithm can efficiently predict and compare many possible routes in time. Though the problem it solves is NP-complete, it is able to reliably mitigate the largest computational bottlenecks while optimizing multiple variables. It performs a flood-type internal simulation and maintains a tree structure which is "pruned" to eliminate redundancy and adapt to network density.

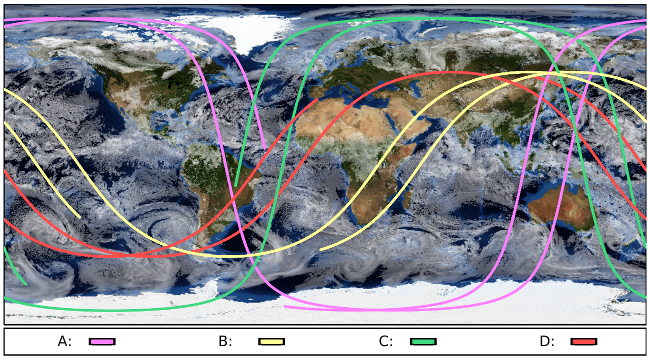
Visualizing and Analyzing Simulation Results

The C++ development library may be selectively configured to produce simulation data. This allows users to capture OSSE behavior in standard data formats. These formats are recognized by visualization and analysis tools in the COLLABORATE Python package. The package provides a collection of high-level executable (and customizable) scripts for animating and plotting network positional data, connections, geophysical data sets, sensor data, communication logs, and system health logs. The executable scripts employ a package of low-level modules (also included) which directly interface to the Python libraries.

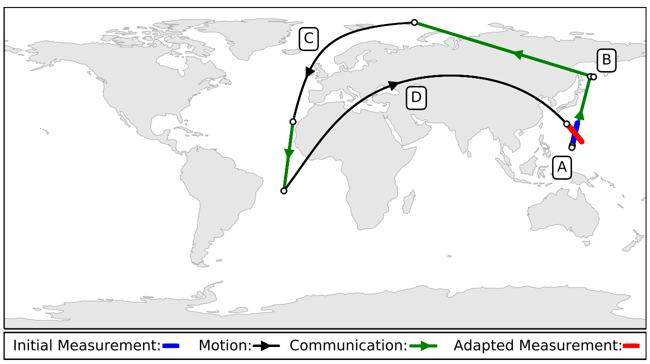
4. COLLABORATIVE SENSING EXAMPLE

An example simulation is presented here to demonstrate the COLLABORATE library features and analysis utilities. This simulation involves an observing system created using the latest TLE sets for common cube satellites (over one hundred satellites total). Only four satellites are active in this example; this example refers to them as satellites A, B, C, and D. Their orbits are illustrated in Figure 3(a). For the example, Satellites A and D are outfitted with cloud radars. All four satellites share a common communication interface, enabling collaboration. Behind the orbits, cloud data is plotted over the Earth's surface. It is provided by NASA through the GEOS5 Nature Run at [7] as total optical cloud depth.

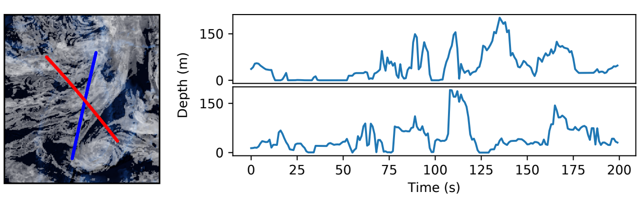
In Figure 3(b), Satellite A is traveling south over the



(a) Orbits of satellites over Nature Run cloud data.



(b) Collaborative sequence - initial measurement, subsequent communication over the network, and an adapted follow-up measurement.



(c) Initial and adapted measurements.

Fig. 3. Collaborative, adaptive sensing example.

Pacific Ocean and is scheduled to obtain a sensor reading (blue line). Internal processing takes place immediately after Satellite A completes its measurement. It is able quantify the measurement's science value and predict the potential science gain and cost of a follow-up measurement by another satellite (decisions made outside the COLLABORATE framework). Satellite A decides that a follow-up measurement is warranted and is able to predict the identity and arrival of the next visitor.

Since Satellite A cannot communicate directly with the next visitor (Satellite D), information must be relayed over the network. The lines in Figure 3(b) illustrates a "collaborative sequence" identified by the communication algorithm described in Section 3. The first transfer is to Satellite B, which holds the information only long enough to transfer again to Satellite C. Satellite C continues in orbit until it can make a reliable transfer to Satellite D over the Atlantic. Then Satellite D adapts its sensor hardware prior to its measurement (red line).

Figure 3(c) shows a comparison of sensor measurements. The map on the left is a magnified view of the target area in the Pacific. The plots on the right show the initial measurement (top) and the adapted measurement (bottom). This simulation's adapted measurement was made possible using COLLABORATE software tools through the collaborative remote sensing model.

5. CONCLUSION

The open-source COLLABORATE software library offers a number of capabilities that could be utilized by future OSSEs. The collaborative sensing model was introduced that captures resource management, collaborative communication, and adaptive sensor reconfiguration. The examples shown demonstrated satellite constellations with realistic sensing platforms, and suggest the value of collaborative sensing networks for expanding the science information provided by Earth remote sensing systems.

6. REFERENCES

[1] Xue, Yong, Li, Yingjie, Guang, Jie, Zhang, Xiaoye and Guo, Jianping, "Small satellite remote sensing and applications - history, current and future", 2008, International Journal of Remote Sensing, 29:15, 4339-4372.

[2] Andre G. C. Guerra, Frederico Franciscoa, Jaime Villateb, Fernando Aguado Ageletc, Orfeu Bertolamia, and Kanna Rajanb, "On Small Satellites for Oceanography: A Survey", 2016, Acta Astronautica.

[3] "Report from the Earth Science Technology Office (ESTO) Advanced Information Systems Technology (AIST) Sensor Web Technology Meeting", February 13-14, 2007, retrieved from *https://esto.nasa.gov/sensorwebmeeting/*.

[4] Hoffman, Ross N. and Atlas, Robert, "Future Observing System Simulation Experiments", *Bulletin of the American Meteorological Society*, vol. 97, no. 9, pp. 1601-1616, 2016.

[5] R. Linnabary, "Collaborate Git Repository," Web, URL: *github.com/ryananan/collaborate*.

[6] A. E. Mitchell, G. E. Smith, K. L. Bell, A. J. Duly and M. Rangaswamy, "Hierarchical fully adaptive radar", in IET Radar, Sonar and Navigation, vol. 12, no. 12, pp. 1371-1379, 12 2018.

[7] "GEOS-5 Nature Run, Ganymed Release." Edited by Steven Pawson and James Gass, NASA, 8 Feb. 2016, Web, URL: *gmao.gsfc.nasa.gov/global_mesoscale/7km-G5NR/*.