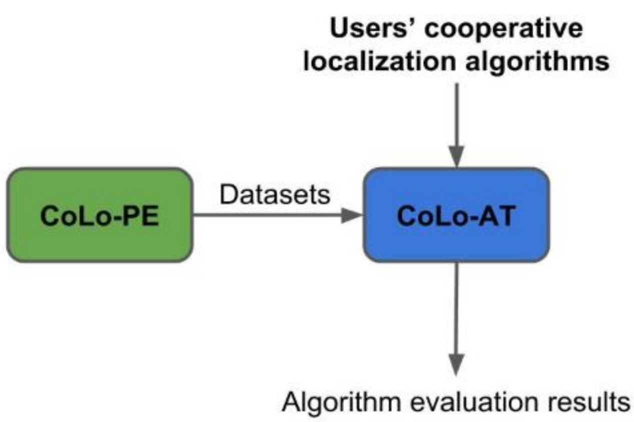
LEMUR Project Proposal

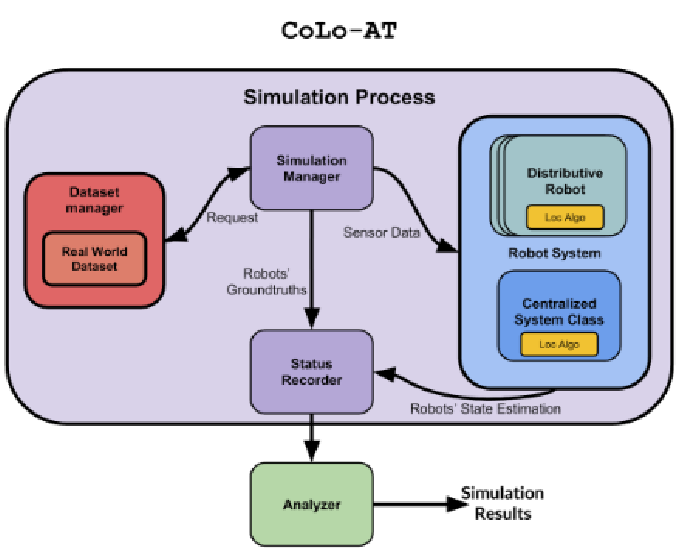
Sagar Doshi Last Updated: April 16, 2019

Big Picture

Cooperative localization aims to solve the problem of accurately estimating the state of a system with incomplete information. Cooperative localization is an area of research interest because of its many applications in transportation, search and rescue, and fully-autonomous robotics[1]. The general approach to solving this problem has been through the development of algorithms utilizing extended Kalman filter (EKF) or Rao-Blackwellized Particle filtering[2] . The current challenge is the improvement and development of algorithms that function in larger-scale implementations and increasingly unregulated environments.

The purpose of Colo is to create a tool kit that quickens the development and testing of cooperative localization algorithms. Colo is unique in that it enables users to test their algorithms on both simulated and real-world data sets. In addition, Colo-AT outputs numerous performance metrics that describe the effectiveness of an algorithm on a given dataset. By providing data collection through Colo-PE and extensive analysis through Colo-AT, Colo enables researchers to more quickly develop or improve cooperative localization algorithms for a broad range of applications.





From Colo-AT Manual

Specific Project Scope

The subset of the cooperative localization problem that is being solved is improving the algorithm development tools available. More specifically, I will extend the simulated data capabilities of Colo-AT and add additional evaluation metrics to its analysis tool. Improving the capabilities of Colo-AT will improve Colo as a development tool for solving the cooperative localization problem.

Colo's code base requires work, so software engineering practices of reviewing documentation and creating detailed specifications will be necessary to improve Colo's functionality. A portion of these practices is dedicated to rigorous software testing to ensure that Colo-AT is augmented and operational without bugs. The specifications and documentation completed during development will be organized to enable future developers and researchers to more easily modify the code base to meet their requirements.

Required Background

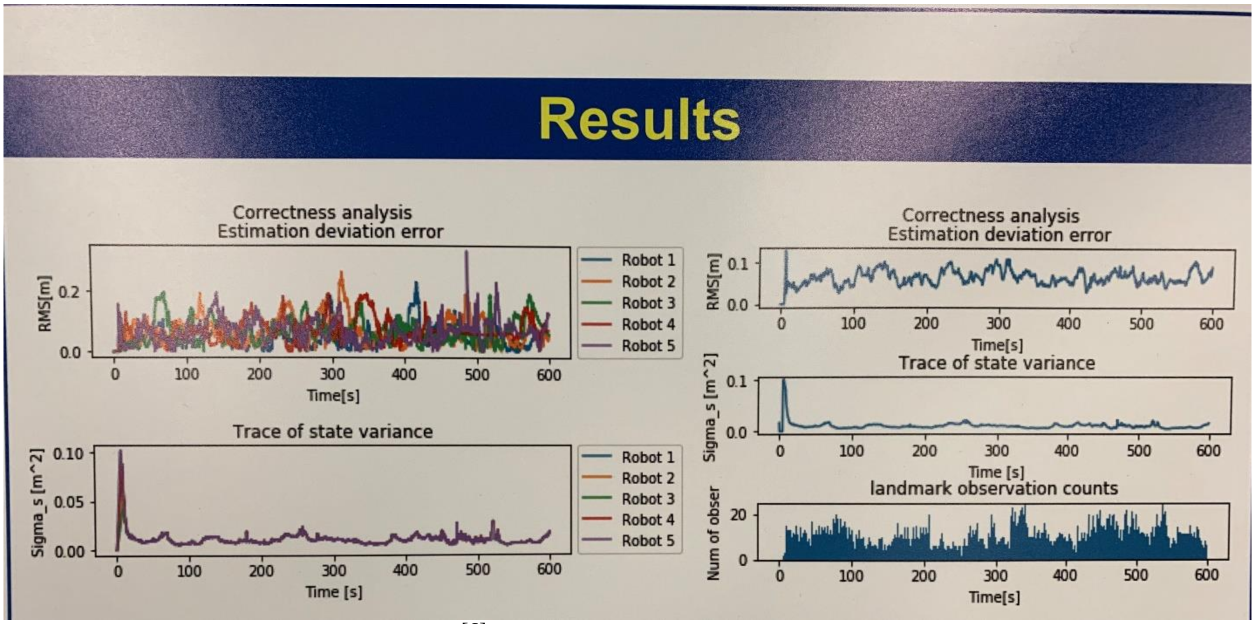
Given that Colo-AT evaluates co-operative localization algorithms, background knowledge of the mathematics involved in cooperative localization is required[3]. In addition, Colo-AT is implemented using Python, so knowledge of the programming language's capabilities and limitations will be necessary in addition to general software engineering skills.

Within Colo is Colo-PE which collects groundtruth, odometry, and measurement data. Currently, Colo-AT has six pre-implemented algorithms the user can evaluate using Colo-PE datasets as well as datasets available on the UTIAS website[4]. The documentation for Colo-AT's current version is only partially complete, and the code base needs to be re-organized.

Related Works and References

When considering the utility of Colo as a performance evaluator, it is important to have a background in the various metrics that be used to judge the effectiveness of an algorithm. General evaluation criteria to be considered when evaluating localization algorithms are scalability, accuracy, resiliency to error and noise, coverage, and cost[5]. Various quantitative measures corresponding to these general criteria can be used to judge the performance a localization algorithm.

Colo was used in a published proceeding from LEMUR[6] to evaluate a new localization algorithm. Colo-AT was successfully utilized to generate correctness analysis via the trace of the state variance and estimation deviation error of the localization algorithm. The plots from the correctness analysis are displayed on the next page.



Results from the proceeding[6] show that the state variance converges within a reasonable range estimation deviation error is also reasonable. They key observation is that neither of these metrics increases over time.

Other available development tools are stage robot simulator[7] and V Rep[8]; they are compared with Colo in the table below (provided courtesy of Colo poster). A key feature to note about these tools is that they are designed for usage with simulated data only, while Colo is robust in that it allows the usage of real-world data as well.

	Purpose	Datasets	Pros	Cons
Stage Project[7]	Multi-Robot Systems	Simulated only	Can be extended to ROS/Stage	Requires specific robot settings
V Rep[8]	General Purpose	Simulated only	Robust Physical Engines	Requires specific robot settings
Colo	Localization Algorithm Evaluation	Both Simulated & Real-world	Easy to set-up Robots	Limited sensors and actuator selection

Goals, Deliverables, Tasks – Spring Quarter 2019

Goals

1. Fully simulated dataset manager that is properly interfaced with the rest of Colo-AT

   Deliverables
   i. Pre-defined path simulation module
   ii. User controlled (keyboard input/window-mouse interface) simulation module

   Tasks
   a. Get simple simulated dataset manager up and running and become familiar with Colo-AT code base
   b. Dynamic data generation within the request response cycle
   c. Handling of multiple robots in the simulation
   d. Add communication between robots to the simulation (with appropriate noise)
   e. Keyboard control module for single robot (for data generation)
   f. (TBD[1]) live tracking/plotting of state and performance metrics using keyboard control

2. Code Clean-up and complete documentation (for both user and development)

3. Power consumption & Communication tracking/analysis

   Deliverables
   i. Module in analyzer/state-recorder/real-word dataset manager to track and plot power usage
   ii. Module in simulated dataset manager to enable tracking of user-defined, theoretical power usage

   Tasks
   a. Edit real-word dataset manager to handle power data
   b. Edit state recorder to handle power data
   c. Edit analyzer to create an appealing/informative display of power data

4. ROS Packages for pre-loaded localization algorithms
   Deliverables
   a. For each pre-loaded algorithm in Colo, a ROS package that can be directly loaded onto Colo-PE robots to physically implement localization
   b. (Optional) – addition of other pre-implemented algorithms into Colo

   Tasks
   i. For each pre-loaded algorithm, a ready-to-load onto Colo-PE ROS package

[1] (TBD) means to be discussed/decided upon

Timeline – Spring Quarter 2019

See above section on Goals/Deliverables/Tasks for more details

Weeks 1&2:

1. Get familiar with Code-base.
2. Get (simplest possible) simulated dataset manager running

Week 3&4:

1. Add dynamic data generation in request/response.
2. Add handling for multiple robots in pre-defined path module (and animated plotting)
3. Begin specifications/outline for keyboard control module
4. Complete the keyboard-controlled simulated module (data generation; only analysis is done by the existing analyzer)

Week 5&6:

1. (TBD) finish real-time plotting/tracking of keyboard-controlled module
2. Code clean-up/documentation & specification sweep

Week 7&8:

1. Begin work on power tracker capabilities of Colo-AT: analyzer/state recorder/real-world-dataset manager modifications
2. Complete Modifications to analyzer/state-recorder/real-word dataset manager
3. Begin implementation of power functionality of simulation

Weeks 9&10:

1. Complete implementation of power functionality in path-defined simulation & user-controlled simulation
2. Start migration from Python → ROS packages for various localization algorithms

References

[1] Saeedi, Sajad, Michael Trentini, Mae Seto, and Howard Li. “Multiple-Robot Simultaneous Localization and Mapping: A Review.” Journal of Field Robotics 33, no. 1 (January 2016): 3–46. https://doi.org/10.1002/rob.21620.

[2] Durrant-Whyte, H., and T. Bailey. “Simultaneous Localization and Mapping: Part I.” *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, 2006, pp. 99–110., doi:10.1109/mra.2006.1638022.

[3] Welch, Greg, and Gary Bishop. "An introduction to the Kalman filter." (2006): 41-95.

[4] Leung K Y K, Halpern Y, Barfoot T D, and Liu H H T. “The UTIAS Multi-Robot Cooperative Localization and Mapping Dataset”. International Journal of Robotics Research, 30(8):969–974, July 2011.

[5] Allen, Michael & Baydere, Sebnem & Gaura, Elena & Kucuk, Gurhan. (2009). Evaluation of Localization Algorithms. 10.4018/978-1-60566-396-8.ch014.

[6] T.-K. Chang, S. Chen, and A. Mehta, “Multirobot Cooperative Localization Algorithm with Explicit Communication and its Topology Analysis,” in *2017 International Symposium on Robotic and Research (ISRR)*, Dec. 2017

[7] Richard Vaughan. "Massively Multiple Robot Simulations in Stage", Swarm Intelligence 2(2-4):189-208, 2008. Springer.

[8] Rohmer, Eric Surya P. N Singh and Marc Freese, “V-REP: A Versatile and Scalable Robot Simulation Framework.” In 2013 *IEEE/RSJ International conference on Intelligent Robots and Systems, 1321-26,* IEEE, 2013, https://doi.org/10.1109/ROS.2013.6696520