

Prototyping a Virtual Robot Swarm with ROS and Gazebo

MATTHEW LAWSON¹ AND GREGOR VON LASZEWSKI^{1,*}

¹ School of Informatics and Computing, Bloomington, IN 47408, U.S.A.

* Corresponding authors: laszewski@gmail.com

project-000, April 30, 2017

Our virtual robot swarm prototype accomplishes a *TBD task* by allocating portions of the task to each virtual robot (VR). As each VR works on its piece of the task, it communicates relevant information back to the master VR. Upon completion, the master VR collates the results and creates a human-readable report. The virtual swarm utilizes the *Robot Operating System* to control the virtual robots, *Gazebo* to simulate the task completion and *RVIZ* to visualize the process. We use *Ansible* to deploy the software to a distributed computing environment. The importance of our effort centers on some super-special conclusion I do not yet grasp.

© 2017 <https://creativecommons.org/licenses/>. The authors verify that the text is not plagiarized.

Keywords: Cloud, I524, ROS, Gazebo, Robot, Swarm

<https://github.com/cloudmesh/sp17-i524/tree/master/project/S17-IO-3010/report/report.pdf>

1. INTRODUCTION

Stating the Problem: Simulating a single robot's actions and responses to its environment prior to real-world deployment mitigates risk and improves results at a relatively low cost. It follows that simulating the actions and responses of a group of robots, e.g., a swarm, will also improve results at a low cost. However, deployment of an interconnected swarm of virtual robots requires much more time and effort than a single virtual robot. Collecting the results from a swarm also requires additional effort.

Our Contribution: We create a cross-platform system to quickly and relatively easily deploy and manage a swarm, as well as evaluate the swarm's operational effectiveness. Or maybe something else...I'm not sure, yet. Automate the deployment of a virtual robot swarm that will accomplish some arbitrary task; capture data from the swarm as it completes its task; report back the results in a human-readable format.

2. VIRTUAL ROBOT SWARM COMPONENTS

2.1. Robot Operating System (ROS) [1]

TBD; will include a discussion of a) how to obtain and install ROS and b) ROS graph concepts. The latter topic will introduce ROS' core components, namely a node, publications, subscriptions, topics and services. This section should probably cover ROS packages and ROS client libraries (primarily C++ and Python)

2.2. Gazebo [2]

TBD; again, introducing core Gazebo concepts. In this case, will include a) world files, b) model files and c) its client-server model. It should also include non-obvious limitation examples, e.g., a gripper arm driven by a single screw instead of multiple screws. In addition, it should allude to any limitations that affect our simulation.

2.3. Ansible

TBD; briefly describe Ansible - what it is, salient features, etc.

2.4. Testing Environment

TBD; briefly describe cloudmesh

3. VIRTUAL ROBOT SWARM PROJECT IMPLEMENTATION

3.1. VR Swarm task

TBD; discuss the task to be accomplished by the swarm, as well as how the information collected during task completion will be communicated back to the master node for collation and reporting.

3.2. Deployment

TBD; document the Ansible steps needed to successfully deploy ROS and Gazebo on multiple computers; will include references for obtaining major components, including adding new repositories if needed.

3.3. Modifications, Pitfalls

TBD; discuss any obstacles encountered with deployment due to dependency problems, connecting ROS and Gazebo, etc.

3.4. Initializing the Swarm

TBD; starting ROS and Gazebo to create the virtual environment; testing swarm interconnectivity; designating master node, etc.

3.5. Begin Task and Monitor Swarm's Progress

TBD; discuss the steps to initiate task completion and monitor the swarm's progress;

3.6. Information Acquired

TBD; discuss the information obtained from the swarm wrt the task at hand as well as each node's vital signs, e.g., battery level;

3.7. Updating Software

TBD; discuss the methods used to implement software updates on each node; remain cognizant of battery levels *I have no idea how I might accomplish an over-the-air update of ROS. This point intimidates me.*

4. VR SWARM PROJECT CONCLUSIONS

TBD; present the data collected in some visualization format; discuss why this project advances robotics forward by utilizing distributed computing;

5. SUPPLEMENTAL MATERIAL

REFERENCES

- [1] Open Source Robotics Foundation, "About ROS," Web page, mar 2017, accessed 16-mar-2017. [Online]. Available: <http://www.ros.org/about-ros/>
- [2] National Instruments, "A Layered Approach to Designing Robot Software," Web page, mar 2017, accessed 18-mar-2017. [Online]. Available: <http://www.ni.com/white-paper/13929/en/>

AUTHOR BIOGRAPHIES

Matthew Lawson received his BSBA, Finance in 1999 from the University of Tennessee, Knoxville. His research interests include data analysis, visualization and behavioral finance.

A. WORK BREAKDOWN

The work on this project was distributed as follows between the authors:

Matthew Lawson. Designed the project in collaboration w/ Gregor von Laszewski, researched the material and implemented the project. Slept far too little.

Gregor von Laszewski. Provided invaluable insights at key points during the process.