Prototyping a Virtual Robot Swarm with ROS and Gazebo

MATTHEW LAWSON[1] AND GREGOR VON LASZEWSKI[1,*]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: laszewski@gmail.com

project-000, March 11, 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. Th importance of our effort centers on some super-special conclusion I do not yet grasp.

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

https://github.com/cloudmesh/classes/blob/master/docs/source/format/report/report.pdf

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.

VIRTUAL ROBOT SWARM COMPONENTS

Robot Operating System (ROS)

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, suscriptions, topics and services. This section should probably cover ROS packages and ROS client libraries (primarily C++ and Python)

Gazebo

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.

Ansible

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

Testing Environment

TBD; briefly describe cloudmesh

VIRTUAL ROBOT SWARM PROJECT IMPLEMENTATION

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.

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.

Modifications, Pitfalls

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

Initializing the Swarm

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

Begin Task and Monitor Swarm's Progress

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

Information Acquired

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

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.*

VR SWARM PROJECT CONCLUSIONS

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

EXAMPLES OF ARTICLE COMPONENTS

The sections below show examples of different article components.

FIGURES AND TABLES

It is not necessary to place figures and tables at the back of the manuscript. Figures and tables should be sized as they are to appear in the final article. Do not include a separate list of figure captions and table titles.

Figures and Tables should be labelled and referenced in the standard way using the `\label{}` and `\ref{}` commands.

Sample Figure

Figure 1 shows an example figure.

Sample Table

Table 1 shows an example table.

Table 1. Shape Functions for Quadratic Line Elements

local node	$\{N\}_m$	$\{\Phi_i\}_m$ $(i = x, y, z)$
$m = 1$	$L_1(2L_1 - 1)$	Φ_{i1}
$m = 2$	$L_2(2L_2 - 1)$	Φ_{i2}
$m = 3$	$L_3 = 4L_1L_2$	Φ_{i3}

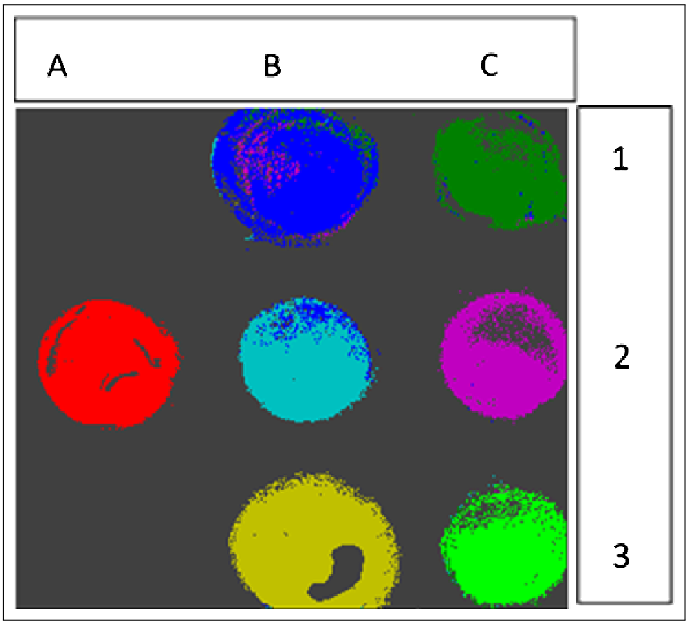


Fig. 1. False-color image, where each pixel is assigned to one of seven reference spectra.

SAMPLE EQUATION

Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed random variables with $\mathrm{E}[X_i] = \mu$ and $\mathrm{Var}[X_i] = \sigma^2 < \infty$, and let

$$S_n = \frac{X_1 + X_2 + \cdots + X_n}{n} = \frac{1}{n}\sum_{i}^{n} X_i \qquad \textbf{(1)}$$

denote their mean. Then as n approaches infinity, the random variables $\sqrt{n}(S_n - \mu)$ converge in distribution to a normal $\mathcal{N}(0, \sigma^2)$.

SAMPLE ALGORITHM

Algorithms can be included using the commands as shown in algorithm 1.

Algorithm 1. Euclid's algorithm

```
1: procedure EUCLID(a, b)          ▷ The g.c.d. of a and b
2:     r ← a mod b
3:     while r ≠ 0 do              ▷ We have the answer if r is 0
4:         a ← b
5:         b ← r
6:         r ← a mod b
7:     return b                    ▷ The gcd is b
```

Algorithm 2. Python example

```
1 for i in range(0,100):
2   print i
```

REFERENCE MANAGEMENT

The best programs to manage your references is jabref or emacs. You can edit the references and verify them with them for format errors. To cite them use the citation key. You can add multiple bib files to the bibliography command separated by comma. Add citations with the cite command. See [1] for an example on how to use multiple clouds. In [2] we list the class content.

Here a test of a citation with an underscore in the url [3].

SUPPLEMENTAL MATERIAL

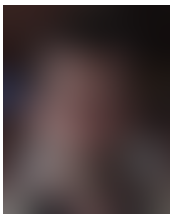
REFERENCES

[1] G. von Laszewski, F. Wang, H. Lee, H. Chen, and G. C. Fox, "Accessing Multiple Clouds with Cloudmesh," in *Proceedings of the 2014 ACM International Workshop on Software-defined Ecosystems*, ser. BigSystem '14. New York, NY, USA: ACM, 2014, pp. 21–28. [Online]. Available: http://doi.acm.org/10.1145/2609441.2609638

[2] Gregor von Laszewski and Badi Abdul-Wahid, "Big Data Classes," Web Page, Indiana University, Jan. 2017. [Online]. Available: https://cloudmesh.github.io/classes/

[3] Web Page. [Online]. Available: http://www.google.com/some_underscore

AUTHOR BIOGRAPHIES



John Smith received his BSc (Mathematics) in 2000 from The University of Maryland. His research interests include lasers and optics.

Alice Smith received her BSc (Mathematics) in 2000 from The University of Maryland. Her research interests also include lasers and optics.

Bruce Wayne received his BSc (Aeronautics) in 2000 from Indiana University. His research interests include lasers and optics.

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.

REPORT CHECKLIST

- □ Have you written the report in word or LaTeX in the specified format?
- □ Have you included the report in github/lab?
- □ Have you specified the names and e-mails of all team members in your report. E.g. the username in Canvas?
- □ Have you included the HID of all team members?
- □ Does the report have the project number added to it?
- □ Have you included all images in native and PDF format in gitlab in the images folder?
- □ Have you added the bibliography file in bibtex format?
- □ Have you submitted an additional page that describes who did what in the project or report?
- □ Have you spellchecked the paper?
- □ Have you made sure you do not plagiarize?
- □ Have you made sure that the important directories are all lower case and have no underscore or space in it?
- □ Have you made sure that all authors have a README.rst in their HID github/lab repository?
- □ Have you made sure that there is a README.rst in the project directory and that it is properly filled out?
- □ Have you put a work breakdown in the document if you worked in a group?