CS491/591: Programming Swarm Robots
Fall 2017

Last changed on November 15, 2017

1 Course Information

Instructor: Matthew Fricke, PhD
Email: mfricke@cs.unm.edu
Website: http://www.cs.unm.edu/~mfricke
Office: Travelstead B09B
Office Hours: Wednesday 3:00-4:00pm or by appointment. Please make appointments by email. There is less likely to be miscommunication that way.

Mailing List:
Subscribe to robotswarm@cs.unm.edu at http://mail.cs.unm.edu.

The email list will serve several purposes. 1) I will make announcements about the class on this list. There is no guarantee that I will also make those announcements in class. I may also make announcements in class that are not repeated on the email list. You are responsible for both. 2) The email list allows class-wide discussion. Feel free to post questions about the material presented in class, the code you are writing for your assignments, and logistic questions related to the class.

Many spam filters will redirect email from mailing lists to a spam folder. You are responsible for making sure you sign up for the mailing list and that you are receiving email from it.

1.1 Course Description

An undergraduate level introduction to selected topics in swarm robotics focusing on the use of robot programming environments such as ARGoS and the robot operating system (ROS) with Gazebo. Topics include a history of robotics emphasising advances in swarm robotics, an introduction to abstract configuration spaces and path planning, sensors and Kalman filters, artificial potential fields, search and resource collection tasks, simultaneous localisation and mapping, an control theory, and swarms as complex systems.

This is the first time this course has been taught. This syllabus is likely to be revised throughout the semester.

1.2 Course Goals

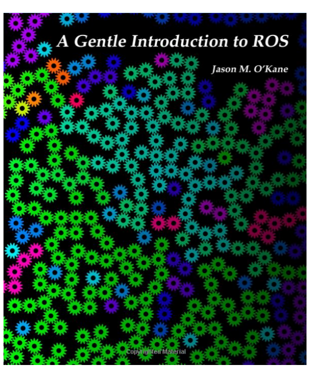
Students enrolled in CS491: Introduction to Programming Swarm Robots learn the foundations of swarm robotics with an emphasis on practical application. In addition graduate students enrolled in CS591: Programming Swarm Robots will read and present seminal papers in the field of swarm robotics. All students will implement significant coding projects to explore these systems. A course emphasis is teaching students how to organise their work into a format suitable for submission to a conference or journal. The projects are presented in a journal style paper and students peer review each others papers.

1.3 Course Format

This is a three credit hour course consisting of two 50 minute lectures per week to be held on Monday and Wednesday, and one 50 minute class per week dedicated to implementation on Friday. The lectures will cover the history of swarm robotics, important algorithms, and background theory. In the weekly implementation class we will discuss specific code issues related to ARGoS, ROS, and Gazebo. You will of course be expected to do the programming for assignments outside of class, but we can try to help each other with difficult technical issues on Fridays. It would be helpful if you could email me issues you would like to discuss in the implementation class the previous Wednesday.

1.3.1 Reading and Software Resources

We don't have a textbook for this course. We will read papers in the field appropriate for an undergraduate class and use other resources such as A Gentle Introduction to Programming ROS, which is a free ebook also available in print from `http://amazon.com/dp/1492143235`.



A Gentle Introduction to ROS
Jason M. O'Kane
CreateSpace Independent Publishing
2013
ISBN-13: 978-1492143239

We will make extensive use of the ARGoS swarm simulator. The ARGoS website contains lots of documentation: http://www.argos-sim.info/.

Further readings will be announced and added here as the semester progresses.

The CS Computer Lab has ROS, Gazebo, and ARGoS installed. You should endeavour to install ARGoS and ROS on your own computer. This will involve installing Ubuntu 16.04 if you do not already have it. The download link is https://www.ubuntu.com/download/desktop.

The Gazebo simulator and AprilTags library are graphically intensive. While virtual machines are a convenient way to setup a linux environment they tend to be very slow. If possible install ROS and Gazebo on hardware with a dedicated graphics card rather than in a vitulaized environment or with an integrated graphics card.

We will use GitHub for all assignments. If you have not used GitHub before you will find this guide useful:
http://blog.udacity.com/2015/06/a-beginners-git-github-tutorial.html

2 Schedule

2.1 Swarm Robotics Background - 3 Lectures

a. Automatons, Androids, and Robots
b. Biological Swarms
c. Major figures in Swarm Robotics

Lecture 1 - Historical Robots (pptx, 1.07 GB)
Lecture 2 - Swarms (pptx, 377 MB)

2.1.1 Getting Started Assignment

a. Sign up for the mailing list.
b. Install ARGoS on your own computer:
   http://www.argos-sim.info/core.php
c. Install MatLab on your computer.
   Instructions for obtaining your UNM license key are here: http://it.unm.edu/download/titles/Matlab-Installation-1608.pdf
   and the software can be downloaded here:
   https://www.mathworks.com

2.2 Embodiment and Control - 4 Lectures, 2 Labs

a. Actuators
b. Sensors

c. Robot geometry
d. PID Controllers
e. Subsumption Architecture
f. The ARGoS environment
g. Homework: Tuning a PID in Matlab

Readings:

i. Intro to Robot Kinematics - Vijay Kumar
ii. A Robust Layered Control System for a Mobile Robot - Rodney Brooks

Videos:

i. Swarmanoid - The Movie - Marco Dorigo

Lecture 3 - Embodiment and Control (pptx, 87 MB)

2.2.1 Control Assignment

Assignment: Description
Code: Download
Groups membership: Link given in class.

2.3 Fundamental Tasks - 9 Lectures

Readings:

i. Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms -Hugh Durrant-Whyte and Tim Bailey
ii. SLAM for Dummies (by Dummies) - Søren Riisgaard and Morten Rufus Blas

a. Associated Project: Robot Navigation and Pattern Formation
b. Shakey the Robot and the A* Algorithm
c. Configuration Space
d. Introduction to Path Planning (Graph, Sampling, and Artificial Potential Fields)
e. Search and Exploration
f. Bug 0, Bug 1, Bug 2, and Tangent Bug
g. Simultaneous Localisation and Mapping (SLAM)
h. Collective Behaviours: Patterns and Communication including Aggregation and Task Allocation

2.3.1 Argos Swarm Assignment

Assignment: Description
Groups membership: Link given in class.

Lecture 4 - Path Planning (pptx, 32 MB)
A* code (htm), (zip with examples, 8 KB)
Artificial Potential Field Code
Lecture 5 - Swarm Intelligence (pptx, 239 MB)
Lecture 6 - Bug Algorithms (pptx, 1.6 MB)
Lecture 7 - Sensor Fusion (pptx, 4.7 MB)
Lecture 8 - Distributed Search (pptx, 21.2 MB)
ARGoS Random Search Code

The ROS Project will make use of all information presented in the next three sections to solve a Central Place Foraging task in the presence of obstacles.

2.4 The Robot Operating System (ROS) - 6 Lectures, 3 Labs

a. Getting Started: roscore and rosrun
b. Event driven systems: Subscribers, Publishers and Topics
c. ROS Packages: The Navigation Stack, Sensors, and AprilTags
d. The TF Node: Coordinate Transformations
e. Kalman filters

2.5 ROS Package Code from Class

Shell Commands
CMakelists.txt
package.xml
randomwalk_node.cpp

2.6 The Gazebo Simulator - 2 Lectures, 1 Lab

a. Building Robots: Actuators, Sensors, Links and Joints.
b. Building the World: Terrain and Obstacles

2.7 The Swarmie Robot - 2 Lectures, 1 Lab

a. A tour of the hardware
b. The Arduino board and software
c. The Swarmie Infrastructure

2.8 Survey of Robot Swarm Systems - 3 Lectures

a. Khepera and Khepera III Robots
b. Alice

c. E-puck
d. S-Bot
e. SwarmBot

2.8.1 ROS Resource Collection Assignment

Assignment: Description

2.9 Final Project Code Development and Testing - 9 Labs

2.10 ROS Presentations - 3 Classes with in-Class Presentations

2.11 Graduate Student Presentations

- Distributed Coordination Algorithms for Mobile Robot Swarms: New Directions and Challenges - Nov 20th (VS)
- From Fireflies to Fault-Tolerant Swarms of Robots - Nov 27th (AA)
- Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making - Nov 22nd (WV)
- Cooperation Between Distributed Agents Through Self-Organisation – Nov 22nd (BD)
- Towards Dependable Swarms and a New Discipline of Swarm Engineering – Nov 27th (PK)
- Exploiting Robotic Swarm Characteristics for Adversarial Subversion in Coverage Tasks – Nov 20th (KR)

3 Assignments and Grading

3.1 Grading

- Matlab Project: 10%, Argos Project: 20%, ROS Project 40%
- Final Exam: 15%
- Class participation: 15% (Attendance, Discussion, Relative Group Contribution)
  In addition graduate students will be asked to give one or more presentations to the class on the papers we read.

3.2 Class Presentation and Response

A 20 minute group presentation.
A Class Presentation Response is required for all students. The responses are factored into the class participation grade.

Please print and bring to class. The forms are due at the end of the class in which the presentation was made.

3.3 Projects

3.3.1 Groups

You will be assigned to work groups. The code and paper will generally receive a group grade unless there is a large disparity in contribution. The reviews are graded individually for each student.

For the ARGoS project Group membership will be determined based on the skills survey from the first day of class. You are free to form your own groups for the ROS project.

Groups will be formed with three members. As the semester progresses students will inevitably drop and be added to the class. Group membership may change during a project. I will try to do this with the least disruption possible.

3.3.2 Project Format

Store your project code in a private GitHub repository. Add your team members and me as collaborators. My github account name is gmfricke.

Use github.com to manage your source code: 35%.
You will be graded on:

1. Commit message quality.
2. Code organisation and readability.
3. Code progression.

Minimum requirements:

1. Each team member must make at least 10 non-trivial commits across at least 7 different days.
2. Github readme describing how to compile and use your code under Ubuntu 16.04. For ARGoS this may include which .xml configuration files to use.
3. 5 page paper: 35%. Use the ACM Latex Template for Overleaf.
4. Paper peer review: 30%, peer review guidelines.

You will be graded on how accurately you review the two papers you are assigned. Project review guidelines are on the course website.

3.3.3 Paper Due Dates

Papers will be downloaded from overleaf and code will be pulled from github on the paper due date specified for each project at 6:00pm Mountain Time.

3.3.4 Paper Review Due Dates

Peer reviews will be due exactly one week after the paper due date.

3.4 Important Semester Dates

https://unmevents.unm.edu/site/academic/

3.5 Exams

Content: there will be two versions of the exams. For undergraduate students the exams will cover material presented in class. For graduate students the exams will also cover the papers listed on the course website. The questions will be designed to test whether students have read and understand the main ideas of the readings. All students will also be expected to be able discuss the ideas and tools they used in writing their projects and homeworks.

Format: short answer.

3.5.1 Final Exam

Friday, December 15th, 7:30-9:30am.

3.5.2 Exam Make-up

There will be a single exam make up time for final exams. The make-up will be on Saturday, May 16th at 8:00am in my office . The make up exam will be different, but at least as difficult, as the regularly scheduled exams.

4 Attendance

- I will not require you to attend class. However, you course participation grade is largely based on in-class activities such as discussion and presentation feedback forms.

5 UNM Resources for Students

- Student Health and Counseling (SHAC): 505.277.3136 (24-hr number)
- Online writing support center: http://caps.unm.edu/services/online-tutoring/online-writing-lab.php

  Please let me know as soon as possible if you require additional support due to disability.