# Orchestration of Intelligent Ground Vehicles into a Homogeneous Swarm via the Google Cloud


Gautam Dash
University of Florida
gbdash@ufl.edu

Joshua N. Weaver
University of Florida
josh.n.weaver@ufl.edu

A. Antonio Arroyo
University of Florida
arroyo@mil.ufl.edu

Sean Wright
Yom Dom Corporation
sean.wright@yomdom.com

Eric M. Schwartz
University of Florida
ems@mil.ufl.edu



## ABSTRACT
In this paper, we describe a Swarm of Robotic Ground Agents, which communicate with each other and collectively solve a myriad of tasks. The system deploys a Base Station, which communicates with all other Agents on the field. The Base Station also communicates with an Operations Center over the Internet. The Operations Center can arbitrate missions and monitor progress through a robust Cloud environment. The Base Station executes orchestration algorithms to achieve mission objectives. It distributes mission parameters to the Agents and relays gathered information back to the Operations Center through the Cloud. We elaborate on the development of our system in a cost effective and efficient manner.



## Keywords
Swarm Ground Robots, Google Cloud


## 1. ROBOTIC PLATFORM

## 1.1 Initial Platform

### *1.1.1 Main Platform*
The original platform was described in detail in our FCRAR 2011 paper [2]. For the current version of each individual robot agent, the XTM Rail 1/8-scale buggy (see Figure 2) was used [1]. The XTM Rail is a remote-controlled racing buggy with four wheel drive and a heavy-duty aluminum chassis and roll cage.


**Figure 2. The XTM Rail.**

The Rail uses a single brushless motor, which is cooled with a large heat sink and two cooling fans. The motor powers three differentials to allow for customization of the vehicle's movements and power output. The Rail is capable of reaching speeds up to 45 miles per hour. Each of the four off-road wheels is supported with threaded aluminum oil-filled shocks (see Figure 3), giving it enhanced stability and durability. The shocks can be adjusted depending on the payload of the vehicle.

The aluminum roll cage is attached to the chassis by six Phillips-head screws and a hinge joint, making it easy to open for access to the interior.

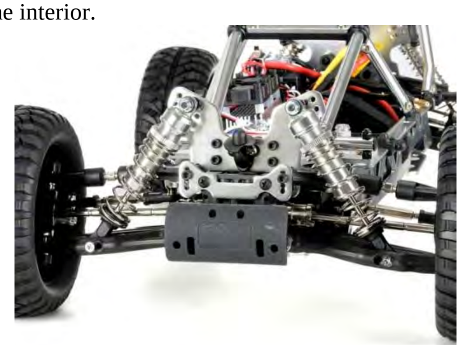

**Figure 3. The front shocks.**

The Rail has a radio frequency receiver to control both the brushless motor and the steering servo. Because of the way the receiver is connected, changing it to be controlled from a computer rather than an external radio signal is a trivial alteration.

**Figure 4. The XTM Rail spoiler and rear shocks. The motor controller can be seen on the right.**

The Rail is powered by a batter pack, and can use NiMH or LiPo packs with voltages ranging from 7.2 to 14.8V.

## 1.2 Updated Platform

### *1.2.1 Additional Equipment*

*Prototype 1:*

The removal of the spoiler leaves four mounting points available on the top of the Rail. A computer case (see Figure 5) was constructed to hold a Linux installation to control the motor driver and steering servo in place of the RF receiver inside the vehicle.

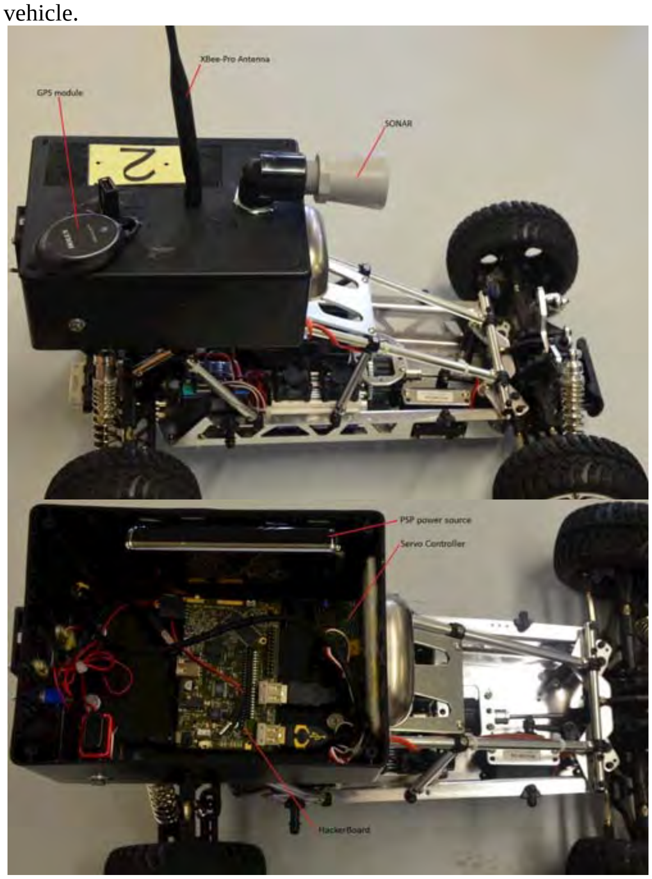


**Figure 5. Prototype 1.**

Inside the case is a Freescale iMX.233 processor board running at 454 MHZ called the HackerBoard [2], which has three USB ports, a standard serial port, and a micro-SD card. The case also contains a power source in the form of a PSP (PlayStation Portable) external power pack. A GPS module and a SONAR system were connected to the processor.

*Prototype 2:*

This version was designed to hold all the electronics inside the roll cage. The computer case has a Texas Instruments OMAP4460 mobile processor board called the PandaBoard [3]. It also has three USB ports, a standard serial port, and a SD card. A GPS module and a SONAR system were connected to the processor. An IMU was also connected to the processor for better estimation of position and orientation.

### *1.2.2 Operating System*

Both the HackerBoard and the PandaBoard have a Linux installation for processing

### *1.2.3 Communications*

One of the USB ports on the board is used to connect an XBee-Pro RF module for wireless communications. The XBee-Pro has a range of 24 kilometers and is used for communication between the robot agents and between each robot and the base station [4].

The PandaBoard also has on board WiFi which and be used to setup an ad hoc network for close range communication. This would enable each robot to communicate with all the others in the Swarm.

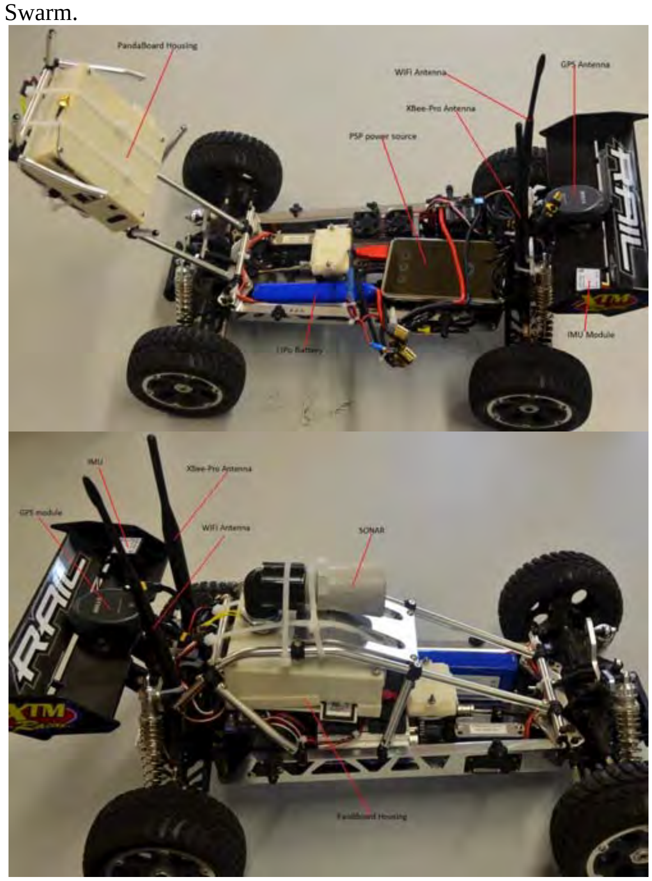


**Figure 6. Prototype 2.**

# 2. SWARM ROBOTS

## 2.1 Definition

The term swarm robotics is used to describe any coordinated multi-robot system made up of individuals which are relatively simple. By working together, a large number of simple robots are able to accomplish more complicated tasks. Swarm robots are often compared to similar swarms of animals, especially insects such as ants, which display similar behavior.

## 2.2 Uses

Swarm robots are notable in that an individual member is very inexpensive compared to a more complicated robot which is able to perform tasks on its own. Because of this, the loss of a single member of the swarm is not a setback in either operations or in cost. If a single multipurpose robot were to be lost while performing a task, it would mean a great loss in both performance and money and time investment. On the other hand, the loss of a single member of a swarm represents a minimal loss of both performance and investment. Individual swarm members can be easily replaced at a cost which is relatively much less than a complicated single robot. In addition, because many of the members of a swarm are the same, the tasks that were being performed by the lost member can be easily done by any other member, resulting in a minimal loss of operational time.

These properties of swarm robots make them suitable for dangerous tasks. If the probability of the loss of a robot during a task is high, then swarm robots will still be able to perform the task without significant loss of time or monetary investment. Since the individual swarm members are relatively cheap and are all able to perform the same tasks as one another, it makes a single unit expendable.

# 3. SWARM IMPLEMENTATION

## 3.1 Implementation

### *3.1.1 Basic Implementation*

The swarm is based on a mixture of placing orchestration algorithms on a base station, mission sets that come from a cloud environment and a robust command interpreter on the individual robot. The individual robots are flexible and programmable with a generic command interpreter that can handle multiple, disparate mission sets.

The mission sets are authorized from the Cloud. The Base Station receives these mission parameters and breaks them down into tasks for each Agent. It then sends them as commands to each Agent using communication packets. It monitors and processes any transmissions received from the Agents and relays analytics to the Cloud.

Each robotic agent has its own execution loop for low level control and self preservation. When it receives any commands, it executes the tasks associated with the command and sends any information processed from sensor data (like position of any obstacles, points of interest) to the Base Station.

### *3.1.2 Software Architecture*

The swarm architecture consists of 3 platforms: Cloud, Swarm Lead, and Swarm Agent. The Cloud platform is a scalable web framework written in Java that communicates with the Swarm Lead using XML protocol. The Swarm Lead is a Java application that is executed on the base station and communicates with one or many Swarm Agents (robots) through communication packets. The Swarm Agent is a C++ application that is executed on each individual robot, which manages motors, sensors and reporting. Each of these platforms follows a Model-View-Controller (MVC) design pattern

### *3.1.3 Basic Execution Loop*

1. A mission is created / authorized in the Cloud.
2. The Swarm Lead detects the authorized mission.
3. The Swarm Lead polls the network for available Swarm Agents.
4. Available Swarm Agents respond to the poll.
5. The Swarm Lead enlists the Swarm Agents.
6. The Swarm Lead spawns Swarm Agent threads and sends mission parameters.
7. The Swarm Agents execute the mission, providing feedback to the Swarm Lead.
8. When mission execution is complete, the Swarm Agents report their mission analytics to the Swarm Lead.
9. The Swarm Lead collects all mission analytics from Swarm Agents, and shuts down the threads.
10. The Swarm Lead reports mission analytics to the Cloud.
11. The Cloud displays mission analytics.

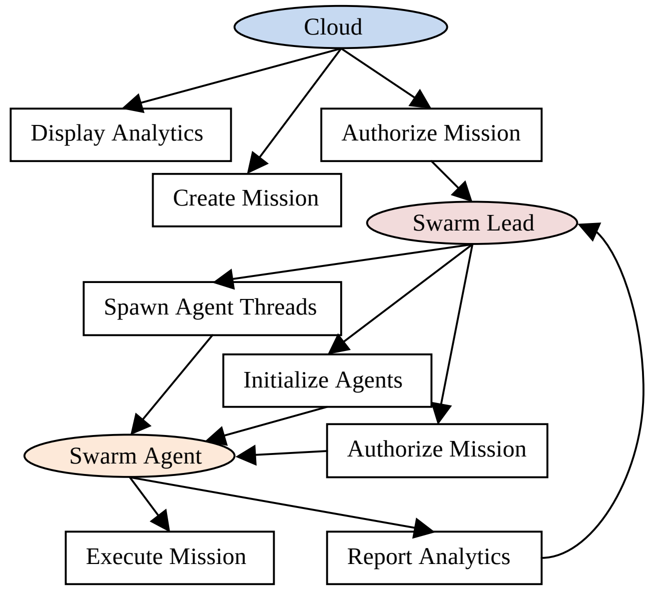


**Figure 7. Process Flow in the Execution Loop.**

### *3.1.4 Recent Changes*

A few problems were identified with the first prototype of the robotic agents. Navigation based on only GPS updates resulted in the Agent overshooting its intended trajectory between GPS updates. To fix this, an Inertial Measurement Unit (IMU) was used to estimate the position and orientation between GPS readings.

The first prototype was slightly top heavy. The second prototype was designed to encase the electronics inside the roll cage.

The second prototype can also use WiFi to create an Ad-hoc network when in range for better bandwidth than serial communication over Xbee.

# 4. SWARM MISSIONS

## 4.1 Search and Rescue

Search and rescue is a basic task performed by the swarm. Like finding the proverbial needle in the haystack, Search and rescue and involves finding a target in a comparatively large region.

Applications of search and rescue include searching for victims in rubble, finding a lost hiker, locating an escaped prisoner, and bomb detection. In many of these cases, the target has a beacon or some other method by which it can be detected. For some of them, such as a hiker's beacon, it may be detected at a distance. For other cases, such as searching for bombs or mines in a field, the detection may only occur at close range by way of a metal detector or chemical sensor. Regardless of the method of detection, the algorithm is similar.

The individual members of the swarm spread out and search for the target signal, whether it is an actual signal or a specific sensor reading. Once a single member is able to find the goal object, it transmits the location's coordinates to the rest of the swarm through the base station.

## 4.2 Area Survey

Area survey is a more specific task in which the swarm robots cover an area, identifying its makeup and creating a survey report. The specifics of the survey can vary depending on the particular task. For example, the swarm robots might perform a geological survey of an area, dividing it into regions of sand, rock, and paved road. As the robots travel over the area, they store data about their current location and the coordinates, creating a map of the area in the process.

Other uses of the area survey task include identification of debris (FOD, or Foreign Object Debris) to determine suitability of a region for various purposes. This can find application in a variety of fields, including construction, cartography, and emergency aircraft landing site identification.

## 4.3 Area Setup

Similar to area survey, the area setup task prepares a surveyed area for use in a specific task. The specifics of area setup are determined by the application. For example, if a region had to be made level for construction work or some other purpose, then debris recognized in the area survey task can be disposed of during the area setup task. This can be accomplished by bringing a specialized robot to clear the debris, either by destroying it or by pushing, collecting, or otherwise relocating it to outside of the region.

The area setup task was successfully applied to a small scale setup of an emergency runway for aircraft. The swarm robots first surveyed the area to determine the most suitable region for a runway, using criteria such as flatness and lack of FOD. They then mapped out a region of the proper size for use as a runway and set up a landing zone. The robots themselves formed the landing zone boundaries and functioned as beacons to mark the region for the aircraft.

# 5. EXPERIMENTAL RESULTS

A Swarm of 5 robotic Agents was used to demonstrate the Area Setup at the Gainesville Raceway. The mission was created and authorized from the Command Center through the Cloud. The Base Station received the Mission parameters through the cloud and tasked individual Agents to various GPS points based on the parameters. Once the Agents were in position, the Base Station relayed the completion status to the Command Center through the Cloud. Figure 8 shows the Command Center Authorization - Arbitration page.

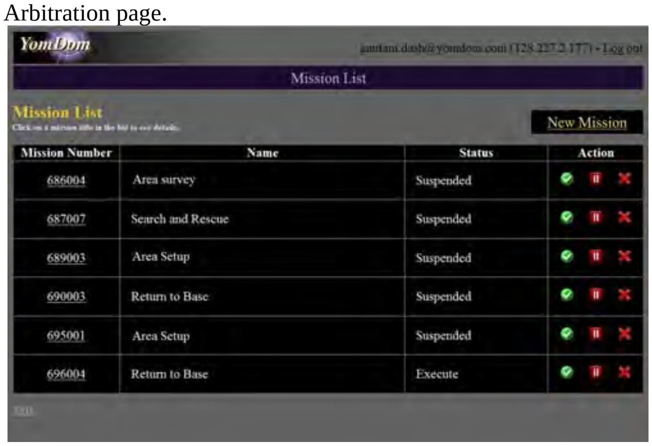


**Figure 8. Mission Authorization / Arbitration**

Each Mission Number on the Mission Authorization page is a link to the Mission Details page which can be used to monitor the results of the Mission. Figure 9 shows a Mission Details page for a 'Return to Base' mission executed at the Reitz Union North Lawn at University of Florida. The two robots used in this experiment acquired their GPS destinations marked '1' and '2' on the Google map. In the Mission shown, they were commanded to return to the base marker (blue '*').


**Figure 9. Mission Details**

# 6. FUTURE IMPROVEMENTS

The current Swarm implementation is based on a Star network topology with the Base Station as the central hub. The Agents are unaware of each other and there only exists a peer to peer communication between the Base Station and the Agents. In the next phase of development, we plan to create more of a mesh topology where in each Agent will be able to communicate with each other and the Base Station.

With a mesh topology, the focus would be to implement algorithms which ensure network connectivity between the Swarm Agents. The challenge here would be to generate trajectories or develop decentralized controllers for the non-holonomic, wheeled robots to transition from one configuration to another [5].

In the current implementation, if an Agent loses connectivity, it tries to return towards the last known location of the Base Station. In future, the Swarm could send another Agent in the last known direction of the lost Agent and the lost Agent could aim to return to the last known direction of the centroid of the Swarm, thus maintaining the efficiency of the Swarm.

# 7. REFERENCES

[1] Team XTM. XTM Racing, 2011. http://www.teamxtm.com

[2] http://wiki.ladyada.net/chumbyhackerboard

[3] http://omappedia.org/wiki/PandaBoard

[4] Emergent Collective Behavior in Autonomous Synergistic Swarm Robots; Sean Wright, Dr. A. Antonio Arroyo, Ian A. Arroyo, Joseph W. Simpkins, Brock Wright, Dr. Lavi M. Zamstein - Florida Conference on Recent Advances in Robotics, Gainesville, FL, 4-5 May 2011

[5] Ensuring Network Connectivity for Nonholonomic Robots During Rendezvous; Z. Kan, A. P. Dani, J. M. Shea, and W. E. Dixon - IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, December 12-15, 2011