**Farm Robotics Challenge – Final Report**

Team Name: **Team Klaatu**
Institution Name: **University of California, Santa Barbara (UCSB)**
Date: 5/14/2023
Authors:

- Undergraduate students: **Jesus Oviedo, Andy Wu, Alejandro Rojas Rodriguez, Aashay Parab, Michael Howo**
- Graduate students and Post-Docs: Liubov Kurafeeva, Animesh Danwal, Tyler Ekaireb, and Bozhang Dong
- Faculty Mentors: Chandra Krintz, Markus Mock, and Rich Wolski
- Contact Information: [jaoviedo@ucsb.edu](mailto:jaoviedo@ucsb.edu), [ckrintz@ucsb.edu](mailto:ckrintz@ucsb.edu)

**1. Problem Statement**

*Describe the farm environment*

We envision a farm environment that is large enough in geographical scope to justify deployment of 1+ robotic platforms including both small scale (tens of acres) and large scale (hundreds of acres and larger) farms supporting a variety of crops and growing conditions. The terrain must be stable and sufficiently level for the robot to traverse; rows should be sufficiently wide for the robot to operate without causing harm to crops or humans. The environment is instrumented with a large number of different sensors that capture information related to soil health, crop health, crop growth stage and rate, input applications (e.g. fertilizer, water, etc.), management strategies, and weather conditions. For scale, the sensors must be inexpensive, resource constrained, and localized (to avoid the installation of electrical power and network infrastructure in the field). Thus they are powered with rechargeable batteries, equipped with low-power storage and Bluetooth Low Energy (BLE) communications, and mounted to a low cost but stable platform (e.g. PVC pipe). Each sensor location and mount point can support multiple sensors and must be accessible by the robot within a few feet of line-of-sight distance.

*Explain farm task/problem, agronomic and business importance, traditional method(s) of addressing the task/problem, and associated difficulty and level of significance – time, safety, labor, expense, etc.*

The problem we address with this project is how to:

- Gather a variety of sensor data from across a farm,
- Provide sensor maintenance (software upgrade, fault testing, etc.) and,
- Deliver electrical power to recharge sensor batteries

for a spatially distributed collection of low-cost, localized sensors located in 1 or more growing blocks on a farm, without the requirement of electrical power or network infrastructure in each block. Sensor data is critical to enable the next generation of data-driven decision support, actuation, control, and automation of farm operations because it provides the basis for understanding and automating management of crop/soil health, enhanced productivity, farm operations, and crop yield prediction. Today sensor data is not collected at all, is not collected pervasively, or is insufficient to drive the next generation of AI/ML algorithms required for advanced digital agriculture solutions. The key reasons for this lack of data support are the high cost of manpower and monetary value to deploy, maintain, and manage even small sensor networks. An additional reason is the cost and invasiveness associated with deploying electrical power and network infrastructure throughout each growing block. Put simply, our approach is to ***bring the infrastructure to each sensor*** via our Amiga robot – named *Gort* – intermittently as the robot circulates through one or more growing regions. In doing so, the project hopes to achieve scale by allowing the sensors (of which there are potentially many) to be as inexpensive and simple (for robustness and maintenance purposes) as possible.

Today, farmers must employ manual labor (and vehicles) to manage sensors (change batteries, (re-)place devices, collect data) or employ low-power long distance radios (LoRa, Zigbee, cellular) which increase device complexity and cost, complicate software development and management, are failure prone in farm settings, and consume significant battery power. By automating farm sensor management with Gort, we can significantly reduce the cost of sensors and data collection (e.g. device cost, labor, cellular services required, etc.), while enhancing the accuracy and effectiveness of data-driven decision support and automation on-farm beyond what is possible today. By doing so, we will lower the barrier to entry for digital agriculture innovation.

Our solution uses Gort to collect sensor data and charge sensor batteries autonomously. The challenges we address are:

- Developing the robot control software to allow Gort to navigate in a growing block so that it visits each sensor.
- Developing and mounting a laser-based charging system and data collection/programming device on the robot that is powered by the robot's on-board power systems
- Interfacing this payload to the robot's driving control so that navigation and sensor interactions are coordinated with robot movement.
- Managing the robot battery life (trip distance, frequency, power use)
- Leveraging computer vision to align robot with the sensor platforms
- Designing and developing a robust software stack for Gort
    - Communication with sensors at each location
    - Measuring health/status of sensors
    - Alerting on sensor failure or other problems that require manual intervention
    - Controlling the laser in response to battery charge level
- Appropriately scheduling data collection and charging in terms of available storage, sensing rate, battery life of each sensor
- Understanding the cost and efficiency of different charging options (what lasers, what light frequency, what photodiode, etc. are most cost and energy efficient)
- Design and implementation of an on-farm data aggregation system from which innovators can extract data to inform and drive new solutions
    - Anonymous data sharing via community co-op "cloud" systems to inform development of the next generation in digital agriculture innovations and products

**2. Societal Impact**

The potential impacts include:

Economic

- We hope that our work, which is at the interaction of mobile computing, AI, IoT, and agriculture, will spur innovation of digital agriculture advances for data-driven solutions.
- Broaden participation in digital agriculture by simplifying large-scale data collection and aggregation
- Reduction of labor costs through better situational awareness and control on-farm without introducing a larger cost associated with the necessary sensing and actuation capabilities
- Extraction of large scale data from a vast diversity of sensors on farms for use in data-driven applications and solutions

Societal

- Individual farms:
    - increase adoption of digital agriculture solutions that enable greater conservation on commercial small and large holder farms
    - lower costs while enhancing sustainability and productivity
- Larger food system:
    - lower the cost of food production and enhance production using the same land
    - reduce the use of chemicals in food production via site specific, precision management made possible by more high quality data

**3. Design Plan**

*As per the rubric document this has been copied exactly as written in the submitted project proposal*

Gort's activities can be divided into three sequential phases. It must

- Identify and locate the next sensor in a list of sensors (ordered by proximity)
- Navigate to the next sensor
- Pause next to each sensor and communicate with the sensor to extract its payload (e.g. temperature data, sensor data, etc.) while the charging system recharges the sensor batteries

For simplicity, we will arrange a set of sensors in a row so that Gort can roll from one sensor to the next and back. These three phases can be decomposed further into

1. Navigation: identification of, and driving to the next sensor, stopping so that the payload is in close proximity to the sensor, and resuming motion once the sensors/power exchange is complete. We envision the use of computer vision to determine location and proximity and telemetry between the sensor and the payload to determine duration.
2. Sensor communication: the payload must gather sensor data, convey any new commands or updates to the sensor, conduct sensor health and status checks, and activate/deactivate the laser (once it is correctly aimed at a photodiode) based on battery charge levels.

3. Power control: aiming and controlling the laser during charging.

We plan to design APIs for these three sets of capabilities (mentioned above) to interoperate as Team Klaatu and then develop them as a subteam using an Agile hardware and software development methodology.

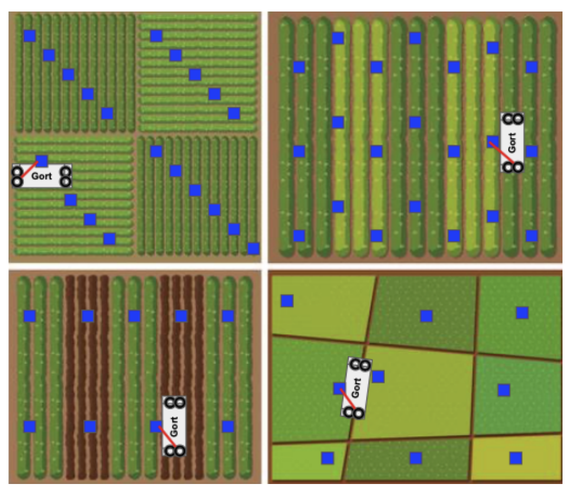
Several factors will affect the performance of the robot. These include:

- Accuracy of locating the sensor using computer vision techniques
- Accuracy of laser aiming so that a small beam will focus on a relatively small photodiode
- Navigation accuracy of the robot with respect to starting, stopping, and holding position during sensor interaction.

From a safety perspective, we have two concerns. The first is with robot motion and, specifically, autonomous control. We plan to incorporate motion and object sensing (using computer vision) along the robot's 1-dimensional path to effect emergency stop in case of an obstruction. The second safety concern is with the use of lasers as a power delivery mechanism. We will begin by using eye-safe lasers (those that operate at a frequency of 1.5 microns or greater) but greater power efficiency needs may dictate the use of higher-frequency lasers. We will adopt these more powerful frequencies when we can assure ourselves that Gort can achieve sufficient proximity between laser and photodiode so as to make an accidental exposure between the two in the "gap" highly unlikely.

**4. Demonstration Guide**

To demonstrate the prototype, our test deployment directs Gort (the mobile charging and data service) to visit and service multiple sensor stations. Gort traverses the field, visiting each sensor station autonomously. Using on-board computer vision running in its payload, it locates each station and directs the motors controlling locomotion to cause Gort to pause at each station it encounters. If data is available or charging is needed, the sensor signals the payload which then connects to the sensor station (via Bluetooth wireless communication) and downloads the sensor data. At the same time, the payload activates a laser that charges the senor batteries via a photodiode mounted with the sensor.




The figure depicts our demonstration design. Gort is depicted as a gray box with wheels. The blue boxes represent the sensor stations deployed across the growing regions; the red line represents the charging laser that Gort uses to charge the station's battery. Gort downloads the data from the sensor station wirelessly.

Our prototype deployment implements the sensor stations using single board computers. We use low power Raspberry Pi Zero (rpizero) devices for the sensors. Each rpizero has a connected temperature and humidity sensor with which it collects microclimate data on a 5-minute duty cycle, and stores this data while it awaits Gort's arrival. In addition, each rpizero is connected to a photodiode that is also connected to a battery charge-controller and an LED. When the laser strikes the photodiode, it begins charging the batter. The rpizero senses the battery charge activity and activates the LED. When charging is complete, the rpizero deactivates the LED.

Each station has an AprilTag that Gort uses to position itself to service the station. We employ AprilTags for their robustness: Compared to the more popular QR tags, AprilTags can be detected from longer ranges, and a wider range of angles. They are also more effective when partially obscured and in low light. They also have high localization accuracy, which allows the payload (implemented using a Raspberry Pi 3) to direct Gort's motor control to adjust its position relative to the sensor station. Gort uses the information to center itself and position the laser to hit the photodiode on the rpizero. The tag also identifies the sensor which Gort uses to contact the sensor for data download. Our representation is implemented using a mutable data structure, allowing for tag and sensor id to be decoupled, even though the marker information is static. Each tag is limited to 12 bits of information allowing for 2^12 sensor stations per tag.

Gort's payload is implemented using a single board computer which is a Raspberry Pi 3B+ (rpi3). We use this device to control and interact with the Gort brain and on-robot devices. We also use it for AprilTag image recognition, communicating with the sensors, downloading their data, and controlling the laser for battery charging (only when Gort is stopped). The rpi3 is also a low power device. We charge it using a PoE switch connected to Gort.

The demonstration video uses a testbed deployed outside our research lab building (Henley Hall) at UCSB. We show images from this demonstration on the right. On the left, we show the prototype sensor station (rpizero, environmental sensors, photodiode, LED, battery, and AprilTag). On the right is a picture of Gort approaching a sensor station. On Gort is an rpi3 and laser in addition to the Gort brain.

With this experimental setup, we showcase Gort's functionalities:

1. Navigational accuracy: Gort moves autonomously to each sensor station with minimal human intervention. To limit navigational errors, we restrict Gort's movement to linear paths within the growing block (crop rows). When passing a sensor station, the rpi3 detects the AprilTag and instructs Gort to slow down. The rpi3 calculates the distance to the center of the marker to enable this. It will tell Gort to halt immediately when the center is detected. We halt Gort by putting it in reverse. If Gort misses the spot, the rpi3 will let it know to adjust and by how much. We have successfully performed this form of autonomy under both day and night time lighting conditions.

2. Sensor station interoperation: Sensors are discovered and detected via the AprilTags. When the robot is centered to the device and fully stopped, it initiates data transfer via bluetooth to rpi3 on Gort. We use bluetooth communication to be power efficient and reduce any interference from other sensors. The collected data includes sensor information and device information. This data is transferred to the Gort brain where our application plots the retrieved data and identifies anomalies to inform the farmer.
3. Power and Laser control: We have successfully implemented laser control for Gort through the rpi3. When Gort detects that it is centered on the AprilTag, it stops and turns on the onboard laser. We also noted the approximate discharge rates for the rpizeros.

**5. User Manual**

Required Parts List:

- Raspberry Pi Zero W for each sensor station desired
- Sensor for desired data acquisition
    - Use of DHT22 humidity and temperature sensor is demonstrated
- Raspberry Pi 3B+ or better (for compatibility with PoE Hat)
- A Raspberry Pi Power-over-Ethernet (PoE) Hat
- Laser Transmitter
- Photodiode/Laser Receiver
- LEDs for each sensor station (optional, for visual feedback of charging)
- Rechargeable Battery Packs for each sensor station
- AprilTags for each sensor station (36h11 tag used in demonstrations)

Hardware Setup:

1. Sensor Platforms:
    a. Connect/Wire the laser receiver and desired sensor to the rpizeros at each platform
    b. Ensure photodiode center is aligned on the platform with the center of its identifying AprilTag
    c. Ensure the platform, specifically the photodiode, is at the same height as the eye safe laser transmitter to be mounted on the Amiga robot.
    d. Keep AprilTag and photodiode as unobstructed as possible to allow for most accurate navigation.
    e. Connect rpizeros to a rechargeable battery pack, specific components are recommended in the parts list.
2. Onboard Amiga Robot:
    a. Securely mount the eye safe laser transmitter on either side of the Amiga robot, at desired height to be matched with height of sensor platform/photodiode.
    b. Insert Pi Cam 2 into rpi3 and securely mount above the eye safe laser transmitter, ensuring the laser transmitter center and camera center align.
    c. Attach Raspberry Pi PoE Hat, ensuring the header pins on the rpi3 are long enough to allow access after having mounted the hat.

d. Wire the eye safe laser transmitter and any control circuit for providing sufficient power to the transmitter from the rpi3.
e. Connect the rpi3 to the PoE switch mounted on board the Amiga robot.

Initial Software Setup:

1. Clone our sensor station and onboard pi repositories that contain setup scripts to install necessary dependencies and robust Python software for our proposed sensor platform and onboard payload (Requires stable internet connection on rpi3s and rpizeros)
2. Upon successful download, run the install scripts designed for their respective platforms
3. On rpi3 and rpizeros, ensure pi/default user has full permissions to dBus/blueZ protocol stack
4. On rpi3, follow Farm-Ng's guide posted on their Discourse forum for setting up a raspberry pi to be configured for the Amiga robot's onboard network.
5. Follow guide provided in raspberry pi repositories for pairing the bluetooth transceiver onboard the rpi3 connected to the Amiga robot to rpizeros to be used on sensor platforms.
    a. Note down the MAC address for each rpizeros bluetooth transceiver, made apparent through the guide.
6. Run the Python software which safely recovers from disconnections. This only needs to be run once to begin data collection. Once the scripts are running, they need not be connected to the Internet.
    a. For rpizeros, the setup may be done prior with a stable Internet connection or connected to over Ethernet for on-field access.
    b. For rpi3:
        i. The setup may be done prior with a stable Internet connection or over Ethernet through the Amiga robot's brain following Farm-Ng's tutorials for how to access the brain.
        ii. Provide sensor station rpizeros MAC addresses along with the identifying tag data (integer 0-X) through a JSON configuration file, further details provided in repository guides.

Gort Application Instructions:

1. Turning the Amiga into Gort is as simple as cloning our application repository into the Amiga robot's app directory by following Farm-Ng's tutorial.
2. Upon successful installation of the app, assuming proper hardware and software setup, to begin data collection a farmer will simply start the app by clicking on it from the home screen.
3. The screen will display the current state of Gort and will await to be given auto control as is necessary for the Amiga to function autonomously.
4. Before letting Gort loose to tend to the sensors, align the robot down the growing block (a linear path) ensuring the sensor platforms are visible to Pi Cam 2 mounting location. Keep Gort as closely aligned to the sensor platforms as possible.
5. During and before operation, farmers may interact with Gort through the app's UI to lower or increase the speed within our set thresholds that allow for precise navigation between sensor platforms.

    a. For night time and low visibility, the speed must be the minimum allowable speed for precise navigation.
6. Though Gort safely halts operation given sufficient obstruction or close person detection, for best operation, keep the current path of the growing block as clear as possible from farmers and terrain obstacles.

Data Retrieval:

1. Gort will automatically retrieve data from sensor platforms once it has reached the target location and aligned itself with the AprilTag.
2. To increase tolerance to failures, data logs retrieved from sensor platforms are replicated on the rpi3 and Amiga, uniquely named with the prefixed with "gort_".
    a. To find the data, on rpi3, "/home/pi/logs".
    b. To find the data, on the Amiga, "/data/home/amiga/logs".
3. A second app has been provided in order to allow farmers on the field to quickly and easily access the collected data. The app parses all on-board data logs into separate datatables for each node. Additionally, the app also provides simple, time-series graphs for easier interpretation while on-field.
4. More datatable tabs and graphs can be added as more sensor platforms are added to the Amiga

Finish and Powering Down:

1. To conclude Gort's current data collection run, safely approach the Amiga robot and exit the app using the Exit button on the Brain screen.
2. Before powering down, ensure proper shutdown of the onboard rpi3 before shutting down the Amiga robot. The Amiga robot is now safe to power down.
3. If needed, ensure proper shutdown of the rpizeros on the sensor platforms before removal and disconnecting from power.

**6. Safety**

Navigation and laser accuracy are at the center of our safety design. We only turn the laser on when the robot is fully stopped and in front of a sensor, so as to avoid potentially harming any people. We also only incorporate eye safe lasers, even the charging laser is designed to be eye safe. In terms of navigation we offload, from the Amiga brain, non-critical computation onto the rpi3. This approach prioritizes object/people detection and emergency stop functions running on the Amiga brain. The emergency stop is based on the people detection api provided by farm-ng and the "braking" is implemented by reversing the direction of the robot's motion and then slowing down to a halt.

**7. Evaluation Plan**

*As per the rubric document this has been copied exactly as written in the submitted project proposal*

*a. Identify 3 metrics to measure the accuracy of your robot and explain their relevance*

1. Navigational accuracy: we will measure the proximity between the payload and each sensor that Gort is able to achieve through navigation. This metric is necessary because the overall effectiveness of the approach relies on Gort's ability to navigate precisely.
2. Energy use: we will measure the energy use of Gort, Gort's payload, the sensors, and the degree of battery charging. This metric is necessary because the energy use will determine the range, and hence the scale, of the approach.
3. Charging efficiency: we will measure the efficiency with which Gort's payload can aim and charge sensor batteries given different sensor duty cycles and activities. This metric is necessary because lasers are an untried technology for power delivery in agricultural settings, particularly for localized battery storage.

b. *Set a data collection goal for each metric (how much data will you collect and why?)*
   i. We will collect gigabytes of image data (still and video) to develop and measure the effectiveness of Gort's navigational capabilities. UCSB has a large-scale private cloud for hosting these data sets and applying off-line analysis that we will use as part of the project. Since our initial attempts at simple navigation will be computer vision based, this data will be necessary.
   ii. We will collect megabytes of telemetry data (both in the lab and on-board) to measure energy usage from the robot, from the payload, and from the sensors. This data is necessary both to optimize sensor communication and power delivery and also understand Gort's range and duty cycle.

**8. Evaluation Results**

The results for our evaluation on our experimental setup are as follows:

1. Navigational accuracy: There are two kinds of accuracy explored here. The first for laser alignment and the second for our safety feature of stopping when detecting a person.
   a. Gort is accurate to about +- 0.25 cm in the day and +-1.25cm at night. This was measured empirically by marking how far from the center of the AprilTag the laser points. This accuracy can be improved by incorporating smarter thresholds. Currently, the thresholds on adjusting Gort's movements are based on the number of times Gort has to adjust to align the laser to the center of the marker. An improvement to this is dynamically adjusting the thresholds by penalizing repeated movements, allowing for more fine-grained adjustments. But if the distance required to adjust is smaller than what Gort can adjust, we'd need to add mobility to the laser itself. This would complicate the design, but allows for more flexible adjustments on Gort's end, overall improving the navigation by increasing error margins.
   b. Our people-detection algorithm provides about 98.67% accuracy. The test data was based on data collected during our experiments of moving Gort around the buildings of the UCSB campus. The ML model used was YOLOv8n and trained on the campus cloud. We selected YOLOv8n for its speed and used it under a low detection threshold to compensate for its comparatively lower accuracy to other YOLO versions, to ensure fewer false negatives.

2. Energy use: We measure energy usage on the rpizeros when connected to a battery pack using powerTop. We note the approximate energy consumed for each mode of operation:
    a. Idle (no computation): 0.311 W
    b. Data Collection: 0.583 W
    c. Data Transfer to Gort(bluetooth): 0.670 W
3. Charging efficiency: Essentially, our charging laser must charge more than the minimum energy consumed during data transfer, or Gort will need to stay at each rpizero for longer. The energy information of the rpizero is also shared during bluetooth transfer and displayed to the user when operating the app.

**9. Research and Design Story**

*Why/how did you select your specific farm challenge/task? (2) Whose expertise did you consult, in and out of your team? What is their specialty and experience? How did their feedback affect your design? (3) What challenges did you face; how did you overcome them, or adapt/pivot your design to accommodate them?*

Our group, RACELab, at UCSB, works extensively with small scale farmers and growers, to build sensor networks, and data collection pipelines, across California. Throughout our experiences we've noticed the recurring problems of utilizing expensive sensors for the main purpose of having additional hardware for better interfacing and networking in technologically deficient farming areas. These hardware peripherals are also what primarily increases the prices of these sensors and make them harder to maintain. Also with any sensor deployment, charging and detecting anomalies in sensor behavior is primarily a manual task, where users would have to go out in the field and replace the nodes, batteries and check for correct data collection.

Based on our own experiences of seeing this problem repeatedly occurring, we designed our solution to tackle these issues, while being generic. Gort is agnostic to the farm location, timing of data collection and allows for low to no human intervention. With its charging capabilities, designed after consulting with the photonics group at UCSB (based on work done in [1]), and interfacing solutions, Gort reduces the amount of manual labor and improves the efficiency required in operating small farms.

This flexibility comes with its own restrictions and challenges. Our design depends on the capability of linear movement and (partially) visible AprilTag markers. We initially tried QR code markers but could not achieve the accuracy and adjustment capabilities provided by AprilTags. Another challenge we faced was navigational control. Originally we kept Gort's capabilities open to include turning and choosing between sensors to collect data from. This quickly became a daunting task, thus we reduced the scope to linear movement only. This also allowed us to focus on safety features and reduce the number of variables to consider when avoiding obstacles.

[1] Nadeem Javed, Ngoc-Luu Nguyen, Syed Farhan Ali Naqvi, and Jinyong Ha, "Long-range wireless optical power transfer system using an EDFA," Opt. Express **30**, 33767-33779 (2022)

**10. Commercial Potential**

*Explain your robot's commercial potential, from a cost and unique competitive landscape standpoint.*

The commercialization potential for the Gort prototype is based on the scaling properties of the approach and the "barrier to entry" into the market that the first mover will erect. Gort is specifically designed to scale the sensing infrastructure. While we have used relatively expensive Raspberry Pi Zero processors in the proof-of-concept, we did so to ease the programming burden and speed the time to solution. A commercialized version would use the same simple meteorological sensors but a low-power microcontroller costing about 2 orders of magnitude less per device to manufacture in bulk. The use of lasers to deliver electricity will create a barrier-to-entry for the competition for two reasons. First, while the solution is well-suited to farm deployment (e.g. it works under canopy, does not require physical contact between devices, is steerable, etc.) getting the necessary energy efficiency will require patentable technologies that will constitute intellectual property. At the same time, an alternative solution based on RF charging (which is currently well-developed technology) is unlikely to be as successful in the farm environment.

Note also that a "mesh" solution to instrumentation at scale inherently scales as the square of the communication devices. With the "data mule" approach that we have explored, there is only a single "expensive" communication component (robot) and the interoperability is bi-lateral (robot to sensor) and not sensor-to-sensor. Thus, our approach enables many more sensors to be deployed on-farm for the same cost (or similarly to employ more advanced sensing and intelligence using the same power budget), and it will greatly simplify the design and development of the next generation of farm sensors – significantly reducing the cost to manufacture, manage, and power them. This scaling also indicates that the operating margins are likely to be attractive. For example, a 10% increase in robot cost is amortized across the total number of sensors a robot can service.

From a market perspective, the next generation of intelligent automation for agriculture requires large numbers of sensors to collect data about the farm, fields, crops, and processes for use in guiding automated actuation and control of farm operations. A large number of sensors are needed because of the inherent variability within the soil, crops, and management techniques across farms.
Decision support, autonomous vehicle operation, and farm automation techniques rely on vast, fine grain datasets to make accurate predictions and safe and effective control. A key reason that precision agriculture adoption has been slow is because of the cost associated with maintaining large sensor network deployments. Using a farm-ng robot to create on-demand scalable infrastructure is likely to unlock this market potential.