Colorado State University

ANT COLONY PARTICLE SWARM OPTIMIZATION AND FORMATION CONTROL

MECH 681A4

PROJECT REPORT

Anil Harish (831009579)

anil.harish@colostate.edu

Shashank Satyanarayana (831070563)

shashank.satyanarayana@colostate.edu

Table of Contents

Abstract

Swarm robotics is one of the ground-breaking methods to test artificial intelligence of ants, bees, wasps, termites and even humans. They possess the ability to function together and perform complex tasks without a centralized control mechanism or high processing power.

Firstly, this report throws light on the biological inspiration of the project and how the behavior and swarming of ants and their colonies dictate and streamline our goal of echoing their abilities of decentralized behavior. We then present the challenges that today's robot systems face to interact with its environment and few drawbacks of single robot systems, also discussing the problems faced by swarm robots few of which we intend to tackle in this project. Later, we present our novel design to achieve two goals in this project (i) Ant Colony Particle Swarm Optimization and (ii) Formation Control autonomously. We also discuss how we can use the same setup for predator evasion functions.

Subsequently, we incorporate IWARD robot standard measuring tools to evaluate or experiments and present our results. Lastly, discuss the techinical difficulties, drawbacks, and future prospects for this system of swarm robots.

Keywords – Swarm; Decentralized; Ant Colony Optimization; IWARD; Autonomous

Contributions

Anil – 50%

Shashank – 50%

Inspiration

Ants: The Eusocial Insects

Ants are one of the most evolved species on earth and are as old as dinosaurs and have been on the Earth for some 200,000 years. Often known as eusocial insects, ants are a taxonomic family that is made up of some 20,000-known species. Current estimates assert that there are about 10 quadrillion ants on the planet. That's 10,000,000,000,000,000, that's at least 10,000 times more than humans. Ant colonies are described as superorganisms because the ants appear to operate as a unified entity, collectively working together to support the colony [1].

Ant societies have: *(i) Division of labour (ii) Communication between individuals (iii) Ability to solve complex problems.* These parallels with the human societies have long been the inspiration and the subject of study.

Behaviour and Ecology

Ants behaviour is described as collective intelligence or "hive" mind. When ants have too many options they make better choices based on the information provided by its neighbours. This avoids information overload and delivers only most important information to the society. This has been intensively studied over the last 20 years [2] to cognition, cooperation and co-ordination abilities of humans.

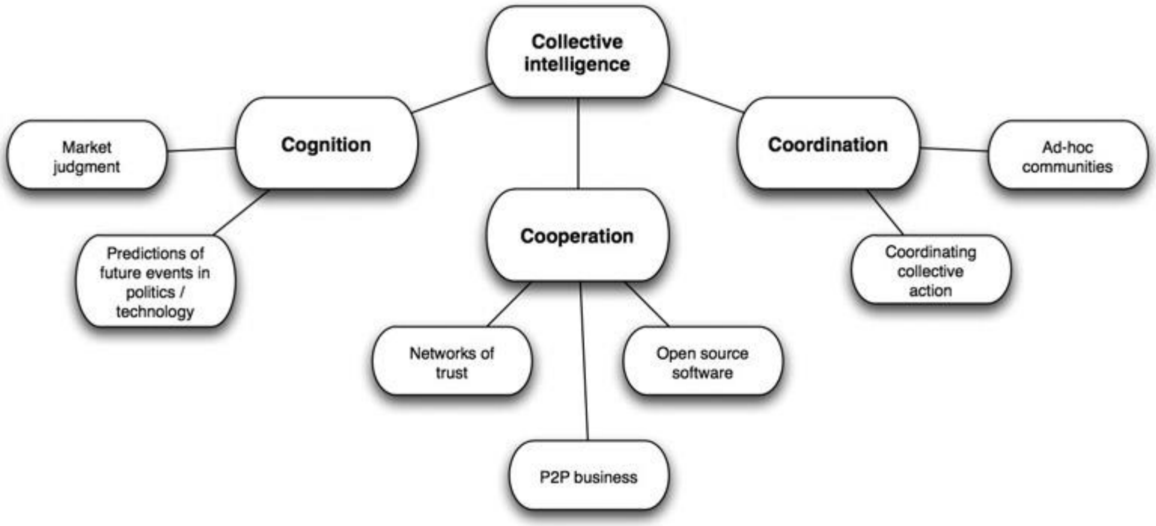


Figure 1 – Collective intelligence studied from ants applied to real world

1. Communication

Ants communicate using pheromones, sounds and touch with the environment and its neighbours. The use of pheromones is highly developed in ants, apart from making trails they can use to communicate organized attack, train (queen ants) and as a propaganda to confuse enemy ants. Ants perceive the environment using two antennae or feelers attached to their head. They can detect chemicals, air currents, vibrations and can transmit and receive information though touch.

2. Defence

Two strong mandibles in the front of the ants can inject formic acid by biting, in some cases stinging into the predator which usually kills them or disables their movement. Soon after army of ants gather around the predator and start eating them even when they are alive.

3. Learning

Many animals have been known to learn behaviours by imitation, but ants are the only species on earth where interactive teaching has been observed. According to Arizona State University [3] controlled experiments with colonies showed that individuals choose their nest roles based on their previous experience and successes.

4. Food Gathering and Forging

Ants are predators, scavengers and indirect herbivores. They direct other ants with pheromones to find food. At first, because it's just one ant trail, the scent is weak – so other ants may wander off that preliminary trail, dropping scent trails of their own. On top of that, the pheromone evaporates quickly. This could get confusing for the ants that follow, but it ultimately works, and here's why: The shortest trails will tend to smell the strongest, because the ant will have taken less time to get home and given less time for the pheromone to fade. That's great, because the strongest-smelling trail will probably be the shortest and easiest trail, and the other ants can make a beeline (so to speak) to the source. So over time, an ordered system emerges out of the ants' chaotic, random search [4].

5. Navigation

Forging ants have the ability to travel 200 meters from their nest and still be able to return home. Distances travelled are measured using an internal pedometer that keeps count of the steps taken and also by evaluating the movement of objects in their visual field (optical flow) [5]. Directions are measured using the position of the sun. They integrate this information to find the shortest route back to their nest. Sometimes they make use of visual landmarks when available as well as olfactory and tactile cues to navigate.

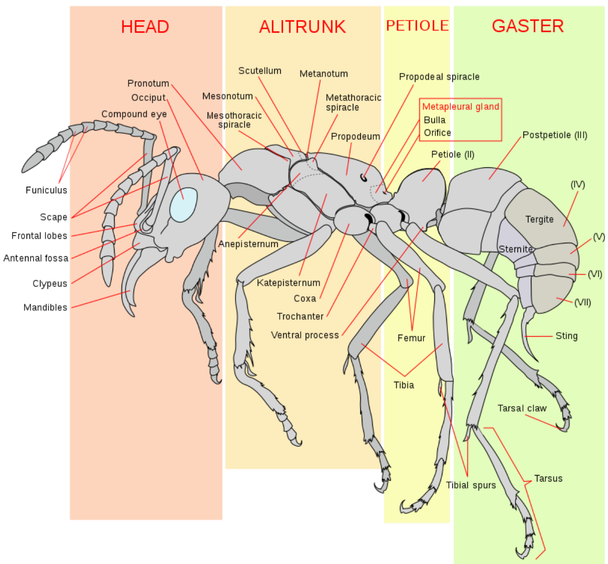


Figure 2 - Diagram of a worker ant (*Pachycondyla verenae*)

Swarm Robotics

Inspired by insects – Mostly Ants

Swarm Robotics is a modus operandi to co-ordination of multi-robot systems which achieve collective behaviour comprising of multi-robot systems which can sense and understand their environment. These robots individually do not have a high computing power. But the system as a whole is able to perform complex tasks and functions which is time consuming. These simple agents interact locally with one another and their environment. Eusociality in these animals and insects give rise to Swarm Intelligence (SI). Swarm Intelligence is the collective behaviour of decentralized, self-organized systems, natural or artificial. The key component of their collective behaviour is their ability to function individually and stronger as a group.

The simplest of mathematical models of animal and insect swarms generally represent individual animals as three following rules:

1. Move in the same direction as your neighbours
2. Remain close to your neighbours
3. Avoid collisions with your neighbours

	Swarm robotics	Multi-robot system	Sensor network	Multi-agent system
Population Size	Variation in great range	Small	Fixed	In a small range
Control	Decentralized and autonomous	Centralized or remote	Centralized or remote	Centralized or hierarchical or network
Homogeneity	Homogeneous	Usually heterogeneous	Homogeneous	Homogeneous or heterogeneous
Flexibility	High	Low	Low	Medium
Scalability	High	Low	Medium	Medium
Environment	Unknown	Known or unknown	Known	Known
Motion	Yes	Yes	No	Rare
Typical applications	Post-disaster relief Military application Dangerous applications	Transportation Sensing Robot football	Surveillance Medical care Environmental protection	Net resources management Distributed control

Table 1 - Comparison of swarm robotics and other systems [8].

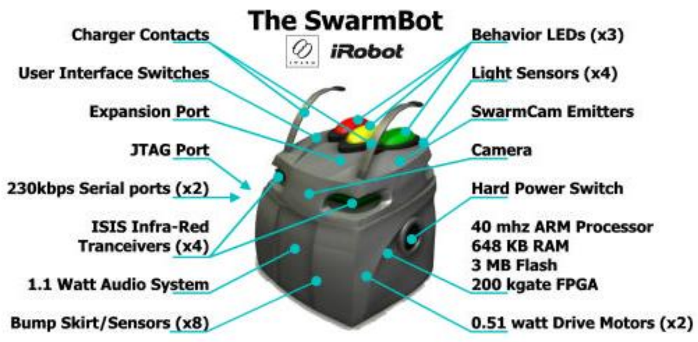
Swarm algorithms follow Lagrangian approach or Eulerian approach [6]. The Eulerian approach views the swarm as a field, working with the density of the swarm and deriving mean field properties. It is a hydrodynamic approach, and can be useful for modelling the overall dynamics of large swarms. However, most models work with the Lagrangian approach, which is an agent-based model following the individual agents (points or particles) that make up the swarm. Individual particle models can follow information on heading and spacing that is lost in the Eulerian approach.

Swarm robots have large number of applications which demand miniaturization such as Nano robotics or micro robotics [7]. Most promising areas include (i) Disaster rescue missions where the robots of different sizes are able to detect presence of life using infra-red sensors and relay the information to a nearby base station for other robots or humans to act (ii) Mining (ii) Autonomous defence functions such as bomb disposal, deter and destroy enemy vessels (iii) cooperative environment monitoring (iv) convoy protection (v) moving target localization and tracking [8].

Leading Swarm Robotics Research Divisions

1. SwarmBot – iRobot/Rice University

RICE university has one of the leading research teams in the field of Swarm robotics. Using real ants as a guide, they have designed and build robot ants with sensors and actuators analogous to their natural counterparts. Their software is written with cooperation in mind, aiming for community behaviours emerging from the interactions of many individuals. Their cubic-inch size produces a robot that is relatively inexpensive and practical to experiment with in a normal-size lab.



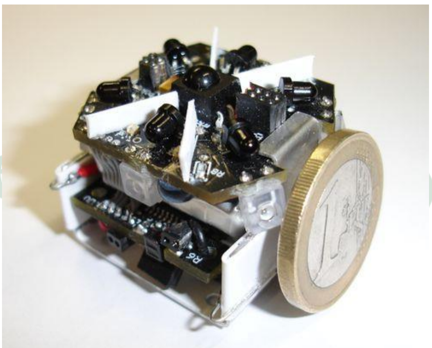


Figure 3 - (Left) SwarmBot - iRobot/Rice University (Right) Jasmine - University of Stuttgart

2. Jasmine – University of Stuttgart

University of Stuttgart, Germany have a research division devoted to development of the open-source hardware and software micro-robotic platform in the size of less-then-3cm-cube [10]. The main goal of this project is to develop a cheap, reliable and swarm-capable micro-robot that can be easily reproduced even at home. This robot allows building a large-scale swarm system (100 and more robots) to investigate artificial self-organization, emergent phenomena, control in large robotic groups and so on. This research is important to understand underlying principle of information and knowledge processing, adaptation and learning for the design and development of very limited autonomous systems.

3. S-Bot – European Commission

One of the main results of the SWARM-BOTS project is the design and physical implementation of 35 s-bots. The s-bots are mobile autonomous robots with the ability to connect to and to disconnect from each other. In addition to a large number of sensors for perceiving the environment, several sensors provide each s-bot with information about physical contacts, efforts, and reactions at the interconnection joints with other s-bots. These include torque sensors on all joints as well as a traction sensor to measure the pulling/pushing forces exerted on the s-bot's turret.

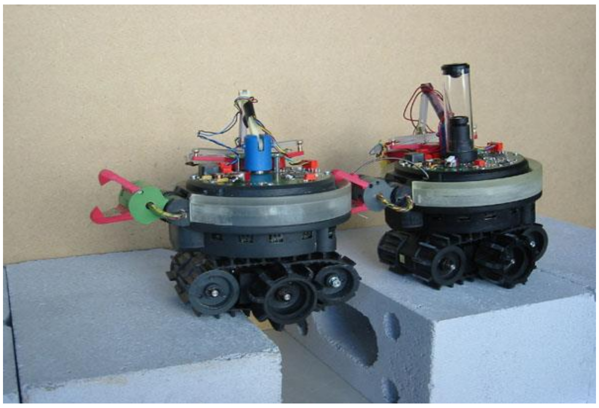


Figure 4. (Left) S-Bot - European Commission (Right) Termes - Harvard University

4. Termes – Harvard University

Inspired by termites, the TERMES robots act independently but collectively. They can carry bricks, build staircases, and then climb them to add bricks to a structure. Harvard's TERMES system demonstrates that collective systems of robots can build complex, three - dimensional structures without the need for any central command or prescribed roles. The TERMES robots can build towers, castles, and pyramids out of foam bricks, autonomously building themselves staircases to reach the higher levels and adding bricks wherever they are needed. In the future, similar robots could lay sandbags in advance of a flood, or perform simple construction tasks on Mars [11].

Dynamic Structure

We modelled each of our individual robots to mimic the different body parts of an ant.
To be precise they do not have the same structure but have similar functions which help us achieve the biological capabilities of the ant and its abilities. In this section we will talk mostly about the hardware parts used in realizing our swarm robot.

Eyes - HC SR04 Ultrasonic Sensor

To resemble the compound eyes and the ant's ability to sense up to 3 meters we used the Ultrasonic ranging module HC - SR04. It provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm.

You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8-cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimetres or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we used over 60ms measurement cycle, in order to prevent trigger signal to the echo signal [12].

Working Voltage	DC 5V
Working Current	15 mA
Working Frequency	40 Hz
Max Range	4 m
Min Range	2 cm
Measuring Angle	15 degrees
Trigger Input Signal	10uS TTL pulse
Echo Output	Signal Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

Table – 2 Electrical Specifications for our Ultrasonic Sensor Design

Figure 5 – Using HC - SR04 sensor to replicate the eyes of an Ant

Antennae - Infrared Receiver Sensor - TSOP38238

We modelled the IR receiver to act like the environment sensory module of the ants. It far from its biological abilities to detect chemicals, air currents, and vibrations. However, it can do the basic and the most important part of communication between the other robots. s. A PIN diode and a preamplifier are assembled on a lead frame, the epoxy package acts as an IR filter. The demodulated output signal can be directly decoded by a microprocessor.

Figure 6. shows the application circuit which is used to realize the sensing of the TSOP38238 sensor along with the Atmega328P microcontroller in order to read the light PWM signals. The Ant's antenna is used to provide a comparison between the biological and actual hardware.

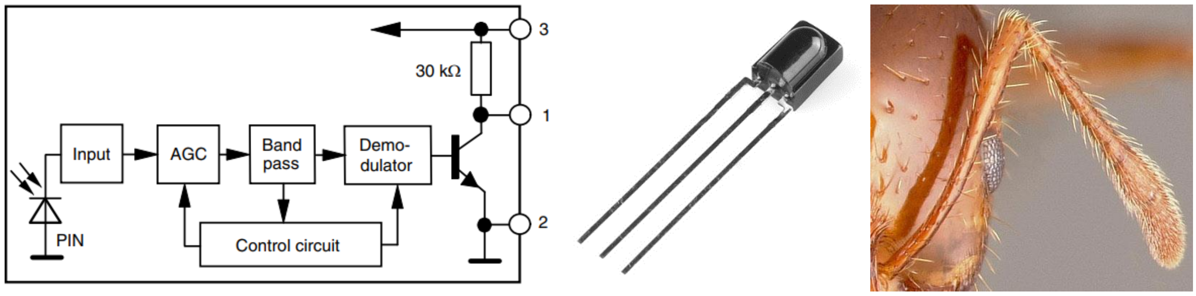


Figure 6 – (Left) Block Diagram of the application circuit (Centre) TSOP38238 IR sensor (Right) Antenna of an Ant

Legs – Wheel

Ants have 6 legs with 3 degrees of freedom which would be very hard to control using a microcontroller and would require 12 servos which will make the project complex by itself. If more powerful Microprocessors are used this would take away the basis of swarm robots having less computing power and intelligence as individual agents. Therefore, decided to stick to a normal wheel.

Body – 3D Printed Chassis

A custom 3 layered chassis was designed for this experiment. The design was compete using AUTCAD and a test model was built using INOVA-1800 3D printing filament, which is a quality co-polyester filament made from Eastman Amphora™ 1800 3D polymer. INOVA-1800 boasts consistent colorant formulation for reliable, repeatable 3D prints. Chroma Strand Labs produces their materials in Colorado, USA. The entire chassis was printed on a LulzBot Mini printer available at the I2P lab at CSU.

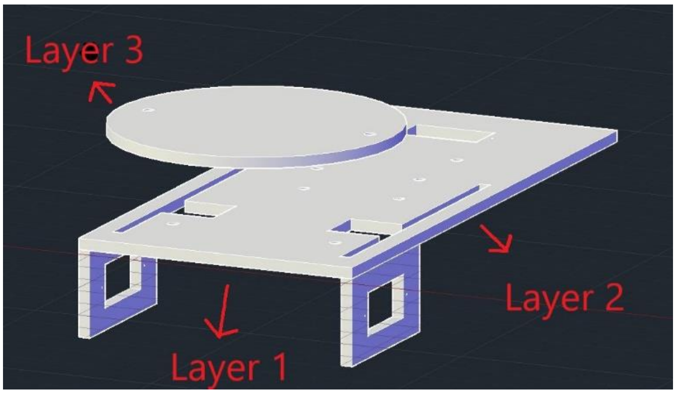


Figure 7 – Peripheral view of the Chassis

Layer 1:

This was the lowermost layer of the chassis, which would house the DC servo motors at the rear end, independently connected to 2-inch wheels for actuation. Special casing between the 2 servo motors would hold the battery securely in position. Additionally, custom holes in front of the chassis were designed for the castor wheel. A 3.5 cm x 5 cm hollow rectangle in the front of the castor wheel to hold the third servo in position to control the direction of the Ultrasonic sensor. Special attention was given to the distance between the castor wheel and the servo to ensure that the movement of the castor wheel would not be impeded by the presence of the servo motor

Layer 2:

The second layer of the robot consisted of custom made sockets to screw the Arduino Uno onto the chassis. Special T-shaped designs around the Arduino board helped to connect wires between the various layers of the bot, conveniently and securely.

Layer 3:

The third layer consisted of a circular 8 cm diameter disc shaped structure, which was placed at 5 cm above layer 2. Special steel holder helped to position the circular disc in position. A 6cm x 6cm breadboard with the IR Transmitters and receivers soldered on it was screwed into position on this circular plate. It was essential to maintain this distance between layer 2 and 3, to ensure that the light-based IR communication between the bots was not impeded by the presence of the forward facing Ultrasonic sensor and its hinge.

Neck – 3D Printed Clamp

A custom clamp was designed to hold the HC SR 04 Ultrasonic sensors in position over the servo motors using AUTOCAD. The clamp was designed such that the two-circular sound emitting and receiving domes were out, while the rest of the electronic parts were securely placed with the hinge to prevent damage to the Ultrasonic sensor in case of a collision between bots. Additional support structures were designed behind the plate to provide strength. Specially designed sockets at the bottom of the clamp helped to glue the arms of the servo motor in position to ensure slippage free movement between the motor and clamp.

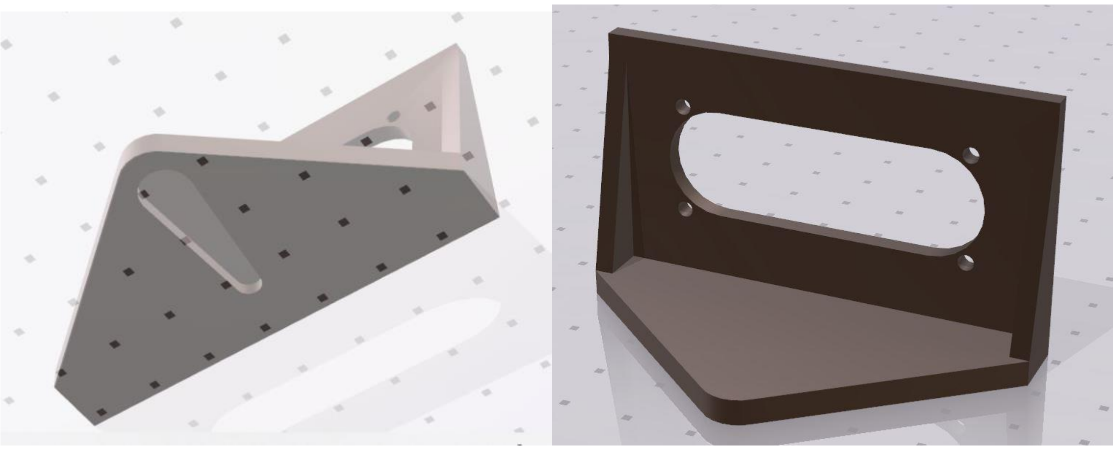


Figure 8 – Clamp (Left) Bottom view shows the slot of servo arm (Right) Back view shows the slot for range finder module.

Pheromones – IR333 5mm 940nm IR LED

For the robots to trace and communicate with the other robots in the vicinity or send broadcasting signals we used the 940 nm ultrabright IR LED for our sensor module. We calibrated the receiving sensor to only pick up wavelength of 940 nm. There are mainly three protocols for this purpose 12-bit, 15-bit and 20-bit for the IR LED. We are currently using the 20-bit commercial grade SONY IR protocol for transmission and receiver decoding [14].

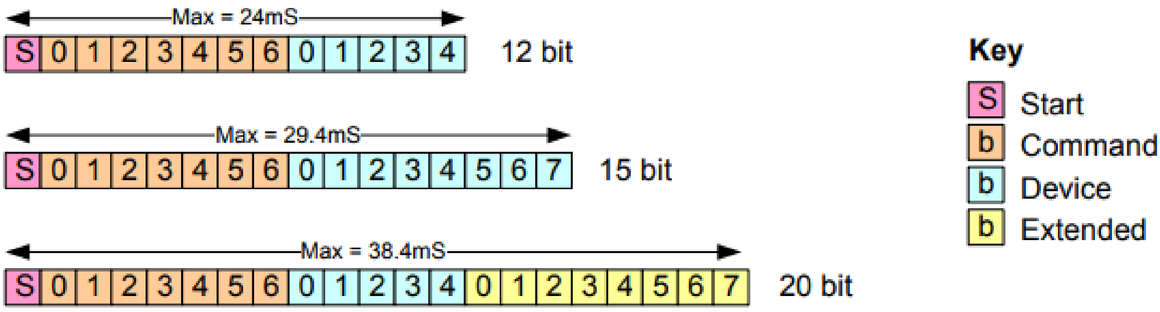


Figure 9. Recalibrated Sony IR protocol

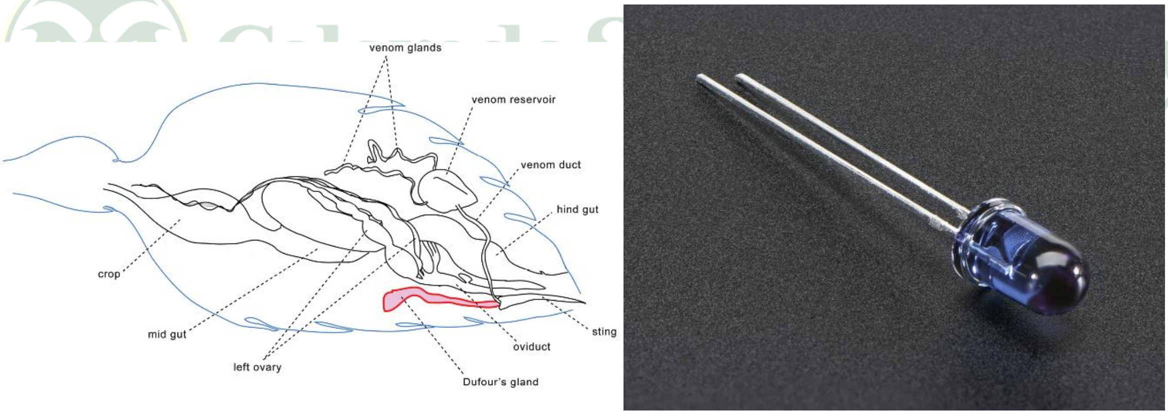


Figure 10. (Left) Pheromone Glands and (Right) IR333 5mm 940nm IR LED

Brain – ATMEGA328P Microprocessor

The Atmel® Pico Power® ATmega328/P is a low-power CMOS 8-bit microcontroller based on the AVR® enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328/P achieves throughputs close to 1MIPS per MHz This empowers us to design the swarm robot module for power consumption versus processing speed.

It boats of 32Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 1Kbytes EEPROM, 2Kbytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), three flexible Timer/Counters with compare modes and PWM, 1 serial programmable USARTs , 1 byte-oriented 2-wire Serial Interface (I2C), a 6-

channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages) , a programmable Watchdog Timer with internal Oscillator, an SPI serial port, and six software selectable power saving modes [15].

The above functions enable us to control a total of 4 IR transmitters, 4 IR receivers, 3 Servos and 1 ultrasonic sensor all in one cycle and one TX/RX port with I2P communication.

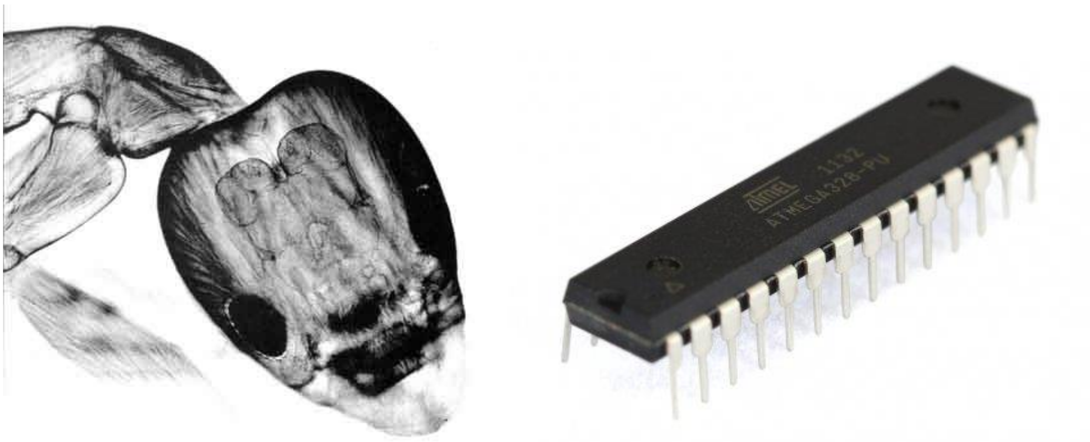


Figure 11 – (Right) X-Ray of an ant’s brain (Left) ATMEGA328/P Microcontroller

Omnidirectional IR Communication Module – Self Fabricated / First of its kind

Our major obstacle in this project was to realise an omnidirectional sensor. This was necessary with utmost importance since there is no sensor available in the market which can sense signals coming from 360 degrees and also make out the difference between those signals at the same time. Nevertheless, modulation and demodulation for transmitting and receiving the signals was another challenge in order to eliminate the interferences from the environment and other IR signals generated by the sun, light emission sources, reflections etc.

The IR-equipment has also the problem of interferences. They appear, like in RF case, when several neighbour robots transmit simultaneously. The problem of IR-interferences can be avoided by restricting an opening angle of a pair IR-receiver-transmitter. For four communication channels, the opening angle of each channel is 90°. In this case we have 2- and 3-robots IR-interferences even in the "closest" radius (50 mm). Reducing the opening angle to 60° or to 40° allows avoiding IR-interferences in the "close" and "near" radius (100 mm) Since many microcontrollers have 8-channel ADC (one ADC input is used by the distance sensor), we choose 4-channel directional communication.

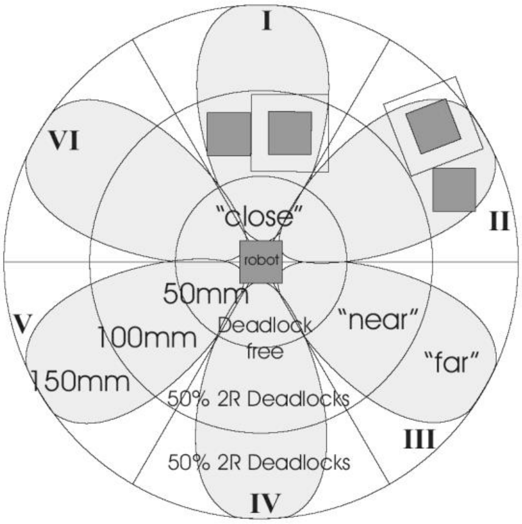


Figure 12 – Initial 6 TX/RX realization (Later reduced to 4 due to overlap)

Total Time required for sharing information for all swam robots

Firstly, we are interested in the number of communication contacts n_c happens during the motion. This value is equal to the average number of robots in the area S_c, n_c=S_c D_sw, where D_sw is the swarm density. We assume that the collision avoiding radius and the robot's rotation radius are small so that we can neglect the area of fractures.

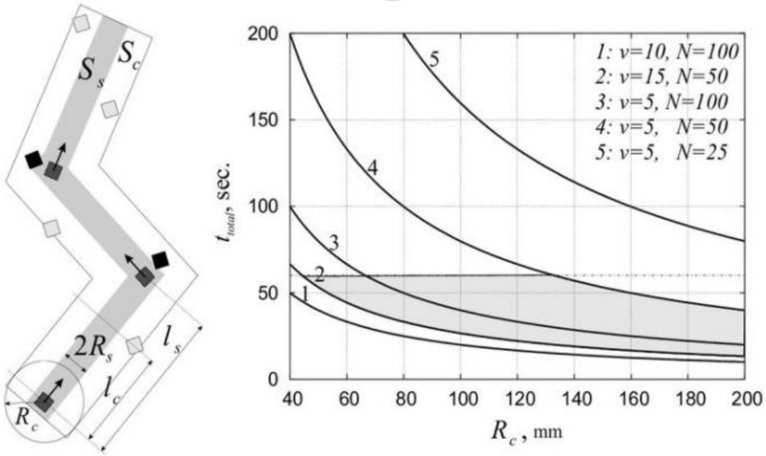


Figure 13 – (Left) Information transmission model (Right) Values Total time required for transmission t_{total} in seconds for different values of N (number of robots)

In this case S_c=2 R_c \upsilon t. D_sw can be calculated as the number of robots N in swarm divided by the area available for the whole swarm S_sw:

$$D_{sw} = \frac{N}{S_{sw}} \quad \rightarrow \quad n_c = \frac{2R_c \upsilon t N}{S_{sw}}$$

From the above equation we can calculate the time till the first infection t_first and the total time t_total= n~t_first+Np_t for infecting the whole swarm as:

$$t_{first} = \frac{S_{sw}}{2\sqrt{2}R_c v N}, \quad t_{total} = N p_t + \frac{S_{sw}}{2\sqrt{2}R_c v N} log_2(N)$$

Where:
nc = Number of Communication Contacts
Dsw = Swarm Density
Rc = Communication radius
v = robot motion velocity
T = time for motor ON
N= Number of robots

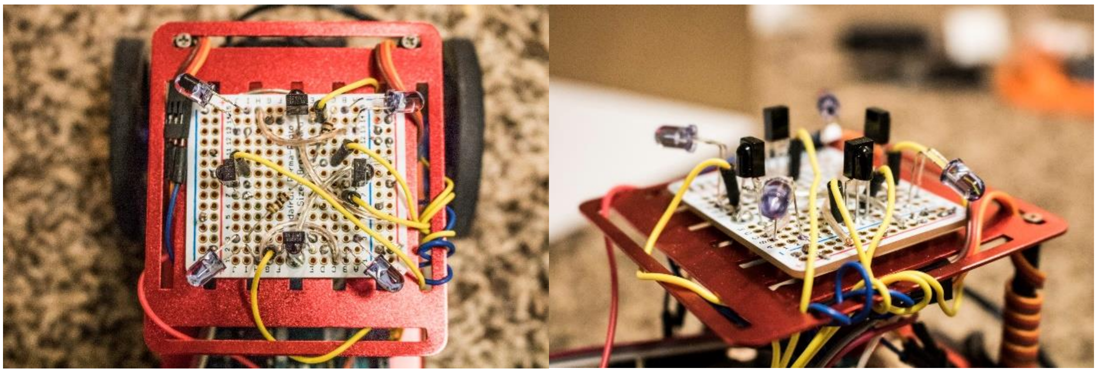


Figure 14. Our fabrication of Omnidirectional IR Transceiver Module

Calculations for our model:

For the considered example with the given S_sw=1000x1000~mm**2, t_total=30~sec., R_c=100~mm and v=20mm/sec. (related to the random motion), the minimal number N_min~ 29 and the critical swarm density D_sw=28.46**(-6). However, in the swarm robotic case with S_robot << S_sw, it can serve as a good approximation.

Developing C++ Library for Omnidirectional IR Sensor

Infrared remote library consists of two parts: `IRsend` transmits IR remote packets, while `IRrecv` receives and decodes an IR message. `IRsend` uses an infrared LED connected to output pin 3. To send a message, call the send method for the desired protocol with the data to send and the number of bits to send. `IRrecv` uses an infrared detector connected to any digital input pin.

The `IRrecv` class performs the decoding, and is initialized with `enableIRIn()`. The `decode()` method is called to see if a code has been received; if so, it returns a nonzero value and puts the results into the `decode_results` structure. Once a code has been decoded, the `resume()` method must be called to resume receiving codes. Note that `decode()` does not block; the sketch can perform other operations while waiting for a code because the codes are received by an interrupt routine.

The interrupt routine is implemented as a state machine. It starts in `STATE_IDLE`, which waits for the gap to end. When a mark is received, it moves to `STATE_MARK` which times the duration of the mark. It then alternates between `STATE_MARK` and `STATE_SPACE` to time marks and spaces. When a space of sufficiently long duration is received, the state moves to `STATE_STOP`, indicating a full transmission is received. The interrupt routine continues to time the gap, but blocks in this state.

The Arena

In order to provide boundaries for our robots to function we developed our own area where we were able to introduce our own obstacles, food source and IR sensor friendly environment for performing the upcoming experiments and record and track the motion of the robots.

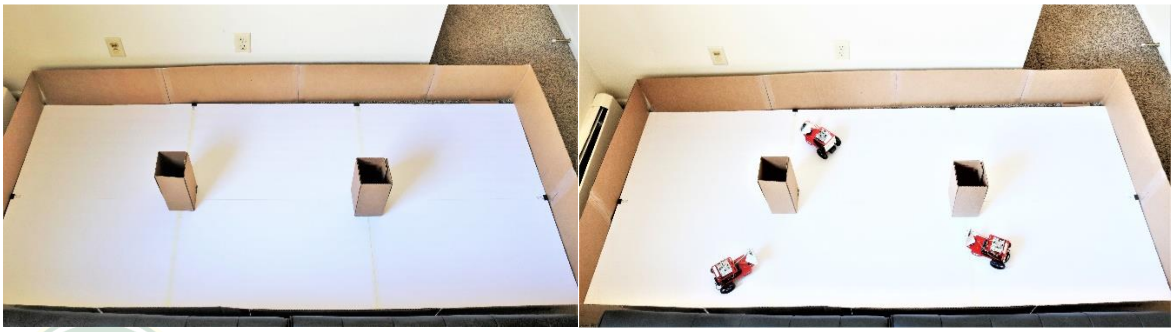


Figure 15 – The Arena and boundaries of operation for the swarm robots

Autonomous Navigation

One of the major ability of any organism is the ability to tackle and navigate itself around the obstacles without any supervision. Ants are also able to perform this with much ease even when they are hanging upside down from the ceiling. We replicated the same ability of the ants using the “Eyes” of the robot - HC SR04 Ultrasonic Sensor.

Figure 14. shows the flowchart and the algorithm used to realize this ability of the swarm robot.

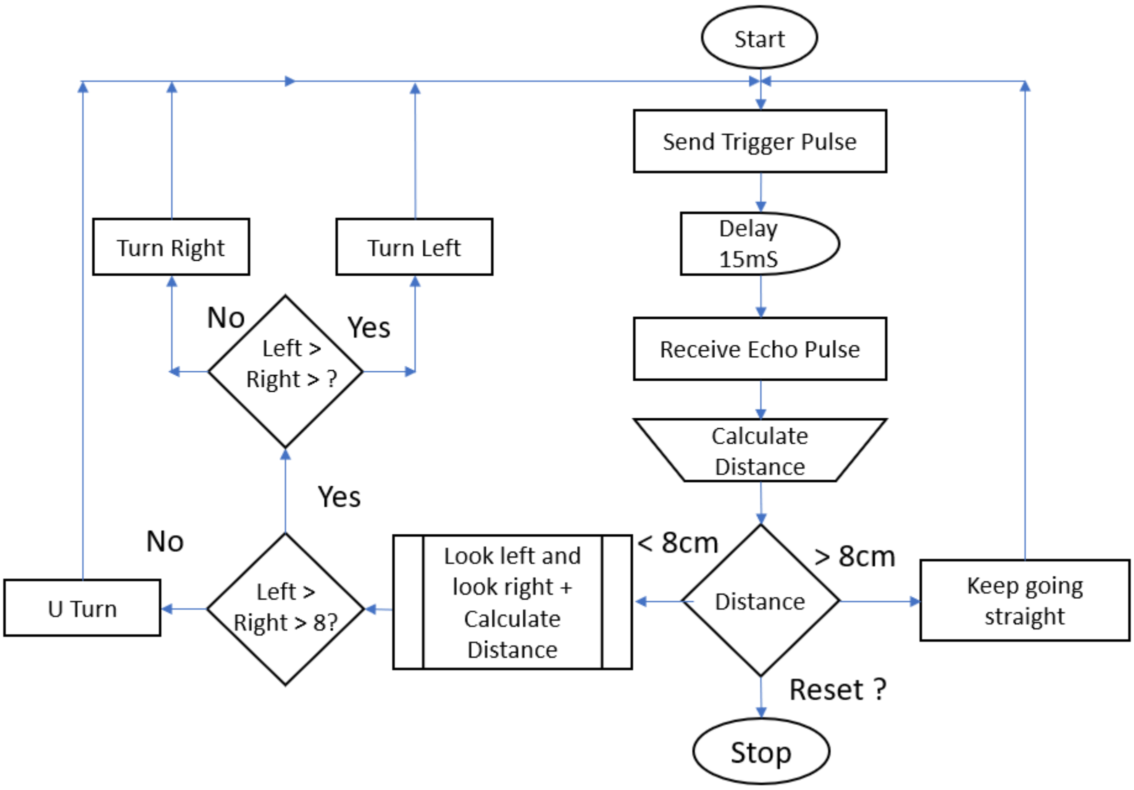


Figure 16. Autonomous Navigation Flowchart

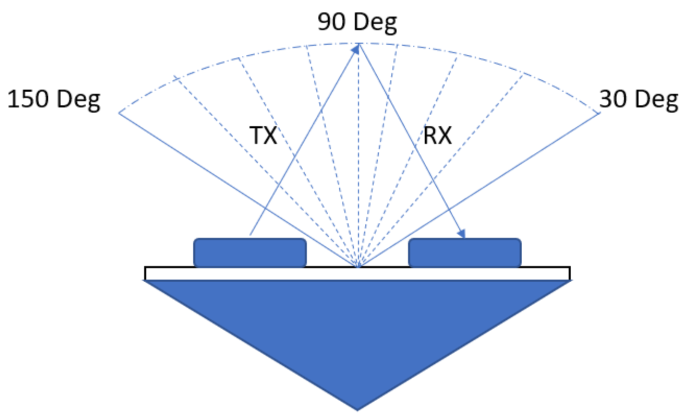


Figure 17. Ultrasonic rangefinder in 1deg steps from 30 deg to 150 deg and calculating distance for every degree.

Artificial intelligence in escaping corners and selecting the best path: Figure 15 shows the unique use of ultrasonic motor for this function. In this algorithm with the use of any servo motor the robot can scout/gauge 150 degrees of its surroundings and not only can it choose the best path but also if it finds itself in a corner it will be able to pull itself out of this corner and start it autonomous navigation again based on the data acquired.

Ant Colony Particle Swarm Optimization Algorithm

Particle swarm optimization is one of the evolutionary algorithms proved to be useful in solving multi-robot tasks. Ant Colony Particle Swarm Optimization (ACPSO) algorithm outperforms other evolutionary algorithms. In this section we will discuss how we can implement the algorithm in boundary environment together with various heuristics improving the performance of each robot and the group. We believe that ACPSO algorithm can solve problems in environments with large area which have similarity to real world problems.

The General ACPSO used for all functions in this project is as follows:

```
for each particle i = 1, ..., S do
   Initialize the particle's position with a uniformly distributed
random vector: xi ~ U (blo, bup)
   Initialize the particle's best-known position to its initial
position: pi ← xi
   if f(pi) < f(g) then
       update the swarm's best-known position: g ← pi
   Initialize the particle's velocity: vi ~ U (-|bup-blo|, |bup-blo|)
while a termination criterion is not met do:
   for each particle i = 1, ..., S do
      for each dimension d = 1, ..., n do
         Pick random numbers: rp, rg ~ U (0,1)
         Update the particle's velocity: vi,d ← ω vi,d + φp rp (pi,d-xi,d) +
φg rg (gd-xi,d)
      Update the particle's position: xi ← xi + vi
      if f(xi) < f(pi) then
         Update the particle's best-known position: pi ← xi
         if f(pi) < f(g) then
            Update the swarm's best-known position: g ← pi
```

Using the above algorithm, we firstly ran MATLAB simulations for 120 ants with communication range of 3 meters to converge on the food particle. Figure 16 shows the simulation at 3 different intervals.

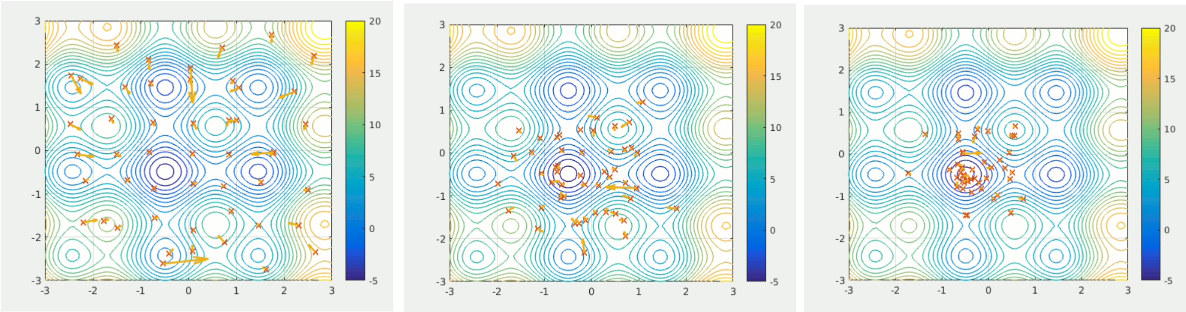


Figure 18. (a) When an agent discovers a solution (b) When information is propagated from one agent to another to swarm to the solution (c) Clustering around the solution

The algorithm works by having a population of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best-known position in the search-space as well as the entire swarm's best-known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

Message Routing

Each swarm robot maintains connections to several "neighbours" in the network, and these neighbouring connections are used for message passing. Suppose that swarm robot X receives a message from Optimus to Bumblebee through robot Y, one of its neighbours. X may have no information clues about where Bumblebee is in the network. However, upon receiving this message, robot X learns something about Optimus: it learns that messages from Optimus come through robot Y. In the future, if node X ever receives a message to Optimus, it can send it back through Robot Y using this clue.

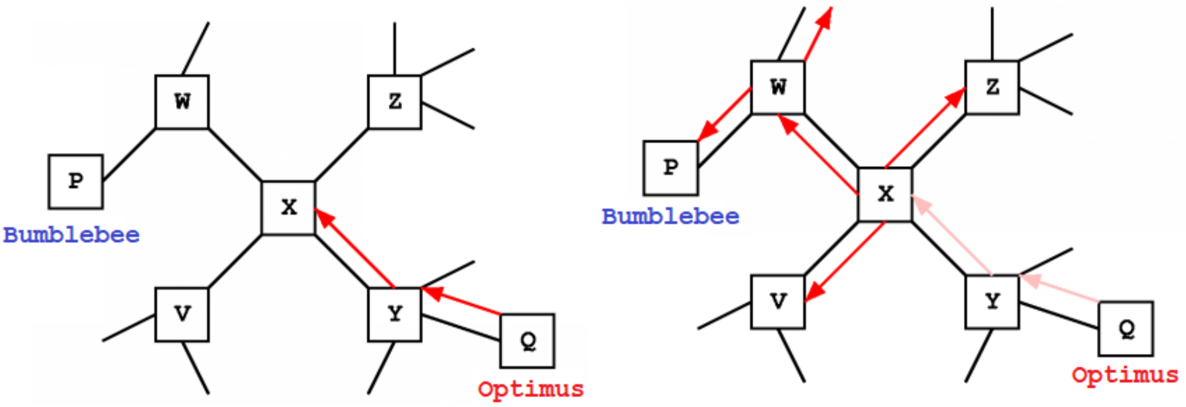


Figure 19. Routing from Optimus to Bumblebee

Regardless of what X learns about Optimus, it still has no information about Bumblebee. The best strategy here, using ants as inspiration, is to "send ants in all directions", or to send a copy of the message on to each one of X's neighbours (what we will call "broadcasting" the message). One of the neighbours may have more information about which direction Bumblebee is in. If none of the robots in the network have clues about Bumblebee's location, they will all broadcast the message to their neighbours. If Bumblebee exists in the network, this technique will eventually find him.

Notice that throughout the search for Bumblebee, the message has been leaving a trail of clues about Optimus. If the message reaches Bumblebee, and then Bumblebee sends back a response, the response can follow these clues on a rather direct path back to Optimus. As shown in figure 18, the response is routed to Optimus, it leaves a trail of clues that can be used to route future messages from Optimus back to Bumblebee. Other robots can make use of these clues too. For example, if robot X sends a message to Bumblebee, the message will travel on a rather direct route using the existing clues.

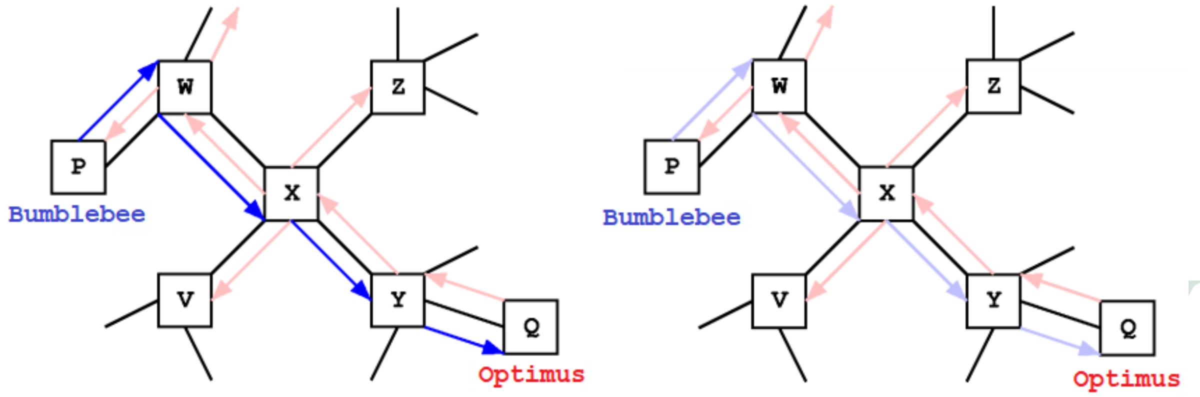


Figure 20. Routing from Optimus to Bumblebee

Infrared Inter-Robot Communication

For four IR sensors and receivers to work coherently, be synchronized and modulate/demodulate the inputs/outputs at the circuit level we had to design and simulate the current and voltage values at end of the communication module. We created a phase lock loop for modulating the output and a band pass filter at the receiver end to segregate the IR signals and other external noise.

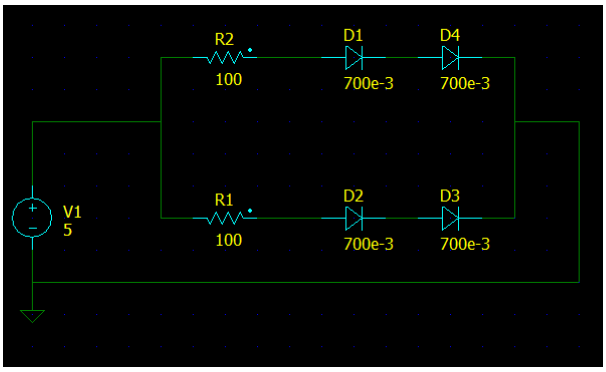


Figure 21. Simple NL5 realization of the mounted IR transmitter circuit

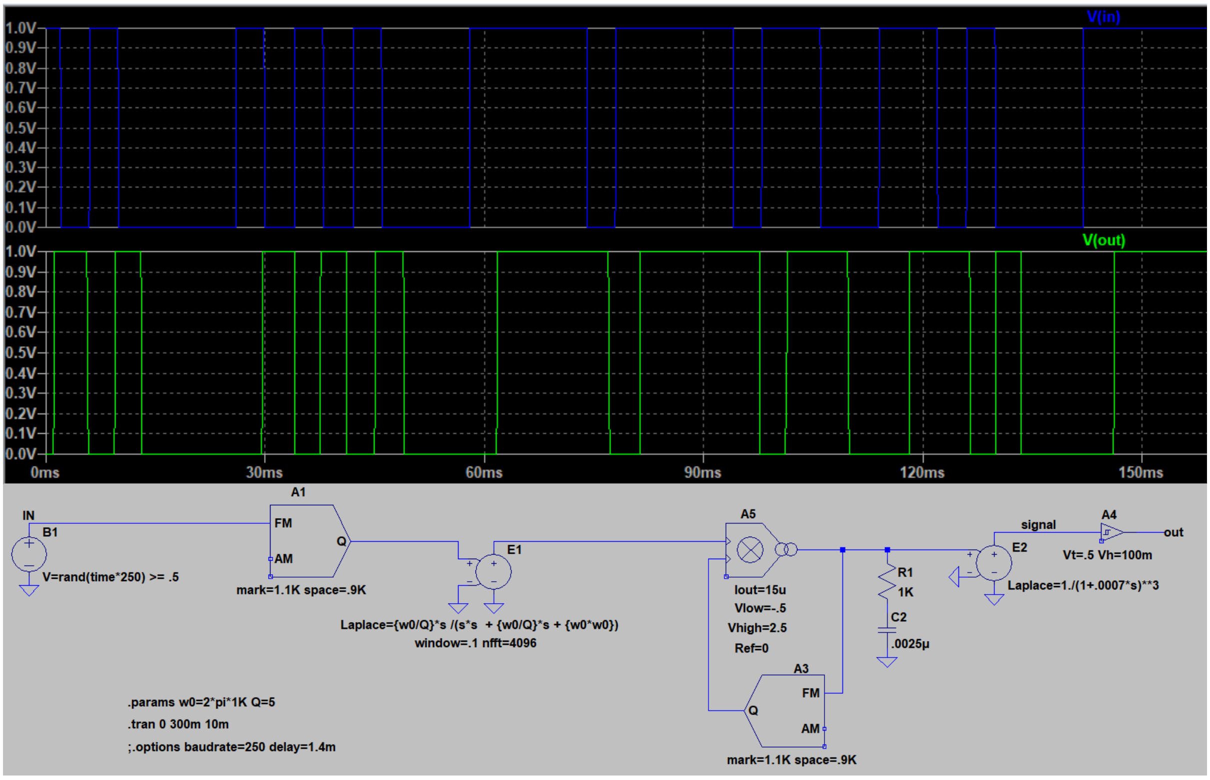


Figure 22. (Top) PWM modulation output for HEX code A90 (Bottom) LTSpice circuit of IR communication

	left	right	delta	min	max	pp	mean	rms	acrms	freq	period	integral
Cursors	300e-3	600e-3	300e-3									
V(D1)	700e-3	700e-3	0	700e-3	700e-3	0	700e-3	700e-3	0			210e-3
V(D2)	700e-3	700e-3	0	700e-3	700e-3	0	700e-3	700e-3	0			210e-3
V(D3)	700e-3	700e-3	0	700e-3	700e-3	0	700e-3	700e-3	0			210e-3
V(D4)	700e-3	700e-3	0	700e-3	700e-3	0	700e-3	700e-3	0			210e-3
V(R1)	3.6	3.6	0	3.6	3.6	0	3.6	3.6	0			1.08
V(R2)	3.6	3.6	0	3.6	3.6	0	3.6	3.6	0			1.08
I(D1)	36e-3	36e-3	0	36e-3	36e-3	0	36e-3	36e-3	0			10.8e-3
I(D2)	36e-3	36e-3	0	36e-3	36e-3	0	36e-3	36e-3	0			10.8e-3
I(D3)	36e-3	36e-3	0	36e-3	36e-3	0	36e-3	36e-3	0			10.8e-3
I(D4)	36e-3	36e-3	0	36e-3	36e-3	0	36e-3	36e-3	0			10.8e-3
I(R1)	36e-3	36e-3	0	36e-3	36e-3	0	36e-3	36e-3	0			10.8e-3
I(R2)	36e-3	36e-3	0	36e-3	36e-3	0	36e-3	36e-3	0			10.8e-3

Table 3. Voltage and current values at each passive device. Notice that clever balancing yields current values to just stay just below the maximum operating current of 30 mA for the IR transmitter

Formation Control

An ant colony consists of about a million ants, mostly travelling in a single file. The surprising nature of this colony formation is that the million ants travelling never face a Traffic Bottleneck or a grid lock. This unique feature of ant colonies can be understood by studying their formation control.

A homogeneous ant behaves according to the given rule and interacts with one another by pheromone. Due to the individual cooperation amongst ants at a micro-scale level, macro-scale foraging behaviour of the colony emerges without any central or hierarchical control. In this section, we aim the study the various formation control mechanisms exhibited by the ant colony and demonstrate them.

An ant colony is typically divided in the structure as shown below. The highest level of a colony consists of Queen ant, which directly monitor a set of Worker ant. Each worker ant then monitors and coordinates the activity of several thousand Scout/Worker Ants.

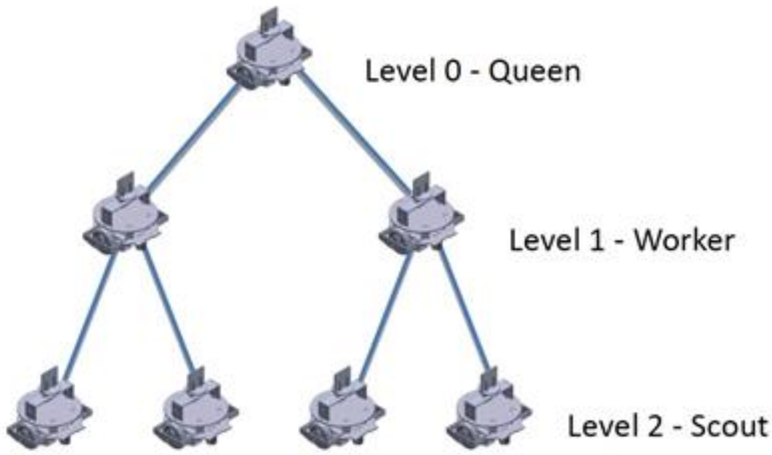


Figure 23. Formation Control Hierarchy

An Ant colony typically exhibits 3 kinds of formation:

Centralized Formation:

This method of consists of a single centralized control unit, with several robots under each unit with its individual ability of perception and execution. This type of formation control is typically exhibited when the food source/target has already been detected and the main aim of the colony is food gathering. Centralized formation control is usually associated with low flexibility and tolerance, since maximum output (food gathering) must be achieved in minimum amount of time.

Layered Formation:

This layer is defined by lack of Centralized control where perception and control abilities are given to the individual ants/bot. There is no central control and individual bots are free to perform their own task according to their will, without reporting to the Scout/Leader while communicating with nearby ants using pheromone. This type of formation control usually has high flexibility and tolerance and is exhibited during food search/randomized walk/exploring the surroundings

Distributed Formation:

This is a hybrid model exhibited by ant colony in which there exists a control unit to supervise the whole system, while the perception control is independent for each robot.
For this project, we demonstrate the Centralized Formation. We use 3 bots to exhibit this formation in which 1 bot acts as the Master with centralized control while the other 2 bots

exhibit the duties of a worker bee. The 2 worker bots contain a PD controller for their movement, which is naturally exhibited by ants in the real world.

PD Controller

A proportional–derivative controller (PD controller or two term controller) is a widely used in control loop feedback system for a variety of robotic applications requiring continuously modulated control. A PD controller continuously calculates an error value as the difference between a desired point and a measured and applies a correction based on proportional, and derivative terms.

In practical terms it automatically applies accurate and responsive correction to a control function. An everyday example is the cruise control on a road vehicle; where external influences such as gradients would cause speed changes, and the driver can alter the desired set speed. The PD algorithm restores the actual speed to the desired speed in the optimum way, without delay or overshoot, by controlling the power output of the vehicle's engine.

For this project we use the following equations:

- Set point – measured distance = error
- Error x Kp = Output for Maneuver
- Output for maneuver = (Distance Set point – Measured Distance) x Kp

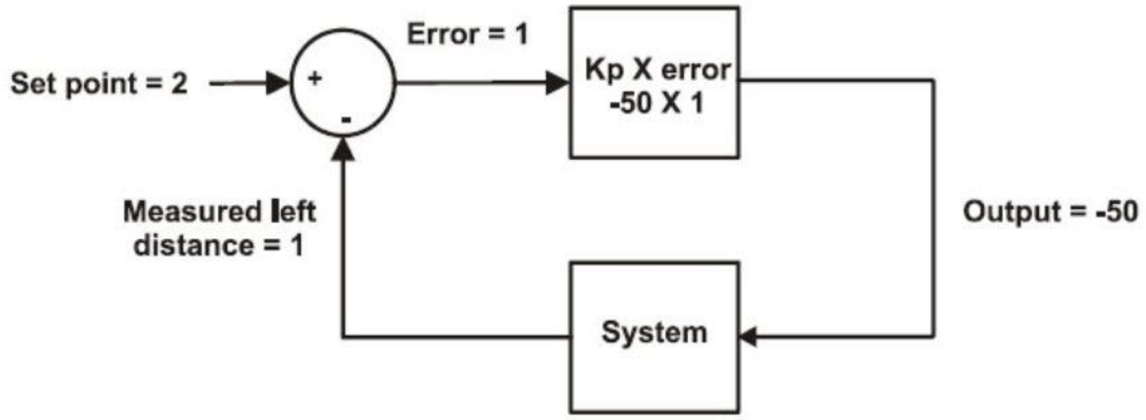
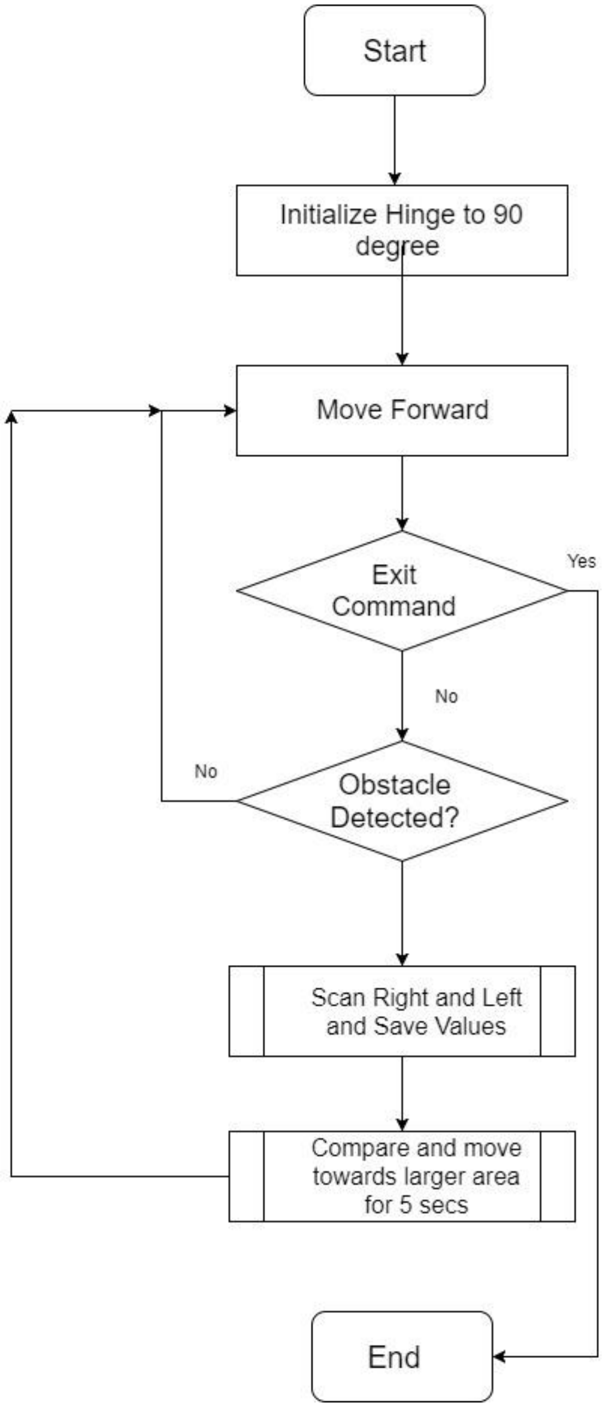


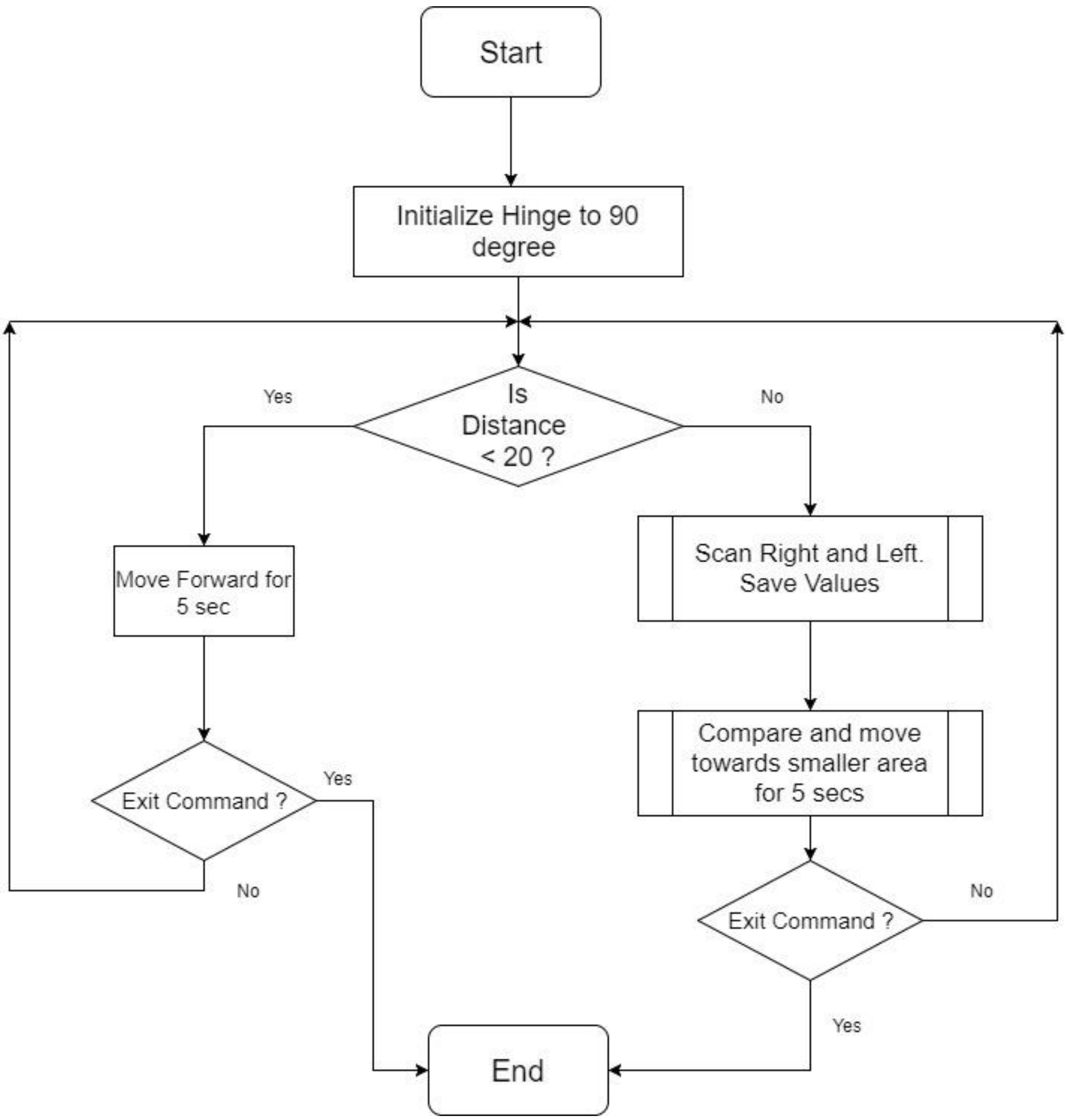
Figure 24. Block Diagram of PD Controller Real-Time Velocity calibration

ALGORITHM

Algorithm for Master Bot



Algorithm for Slave Bot



Results

The following figure shows the performance of the PD controller which we developed:

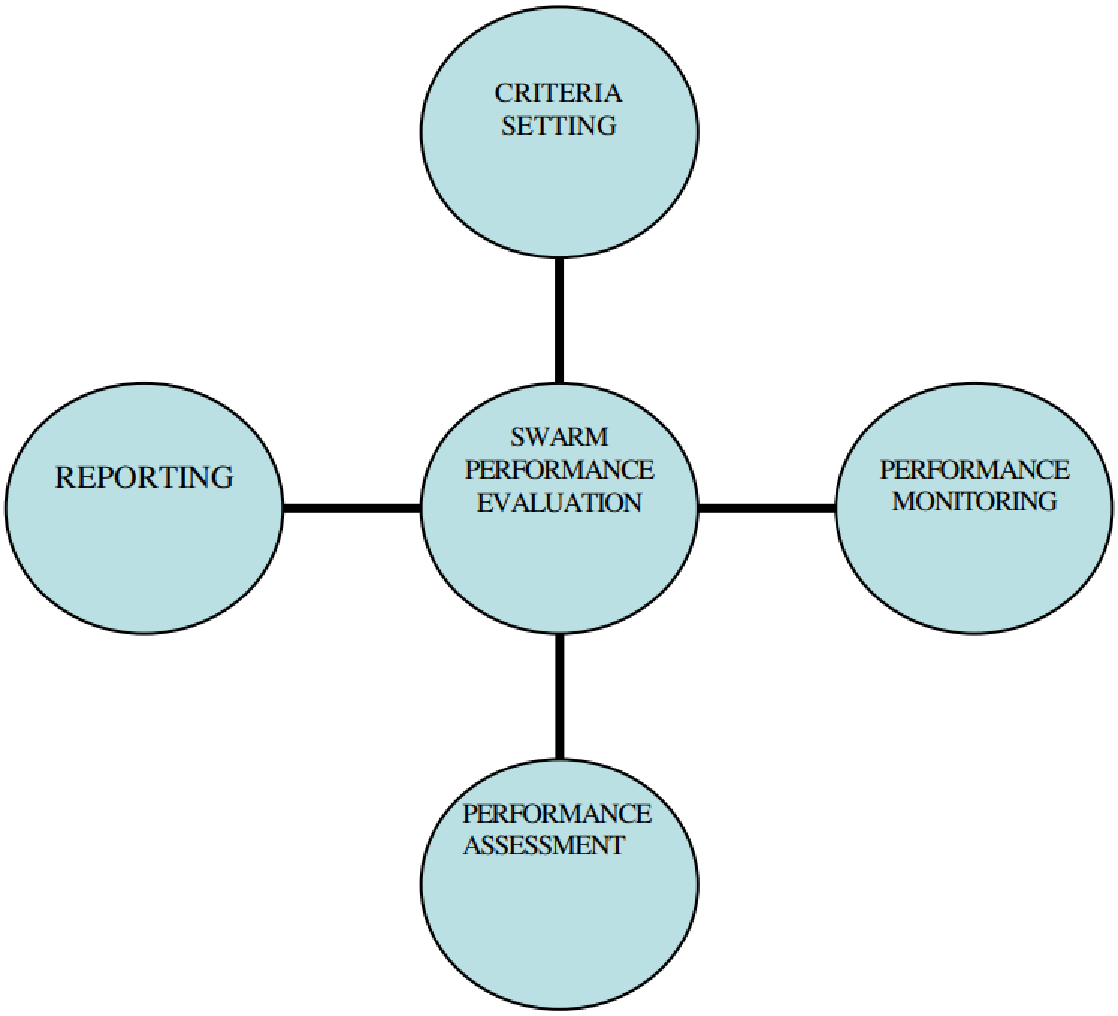
Condition	Measured Distance	Set Point	Error (set point–measured)	Output (Kp × error)	Maneuver Result
	0	2			
Too close	1	2	2–1=1	2×–50=–50	Slow reverse
Just right	2	2			
	3	2			
Way too far	4	2	2–4=–2	–2×–50=–100	Fast forward
	5	2			

Table 4. PD Controller Outputs

In this project we presented a framework for controlling mobile multiple robots, which communicate using IR communication. Mobile and static agents collect the coordinates of scattered master/scout robot and simulate the ant colony formation control algorithm.
Since our control system is composed by several small static and mobile agents, it shows an excellent scalability. Additionally, the use of IR sensors decreases the amount of the necessary communications. They make mobile multi-robot applications possible in remote site with unreliable communication or intermittent communication. Although out primary application is ant colony formation control, the system should have a wide variety of applications.
We have implemented a team of 3 mobile robots to show the feasibility of our model. Even though the robots have basic collision avoidance mechanism, they do not have computing capabilities to determine the exact positions they should stay when they get together to make clusters while avoiding collision. Current implementation is just for exploring the basic behaviours of multiple robots.

Evaluation of the functions at IEEE Standards

This system is mainly developed for IWARD and European 6th Framework Research Program developed for Intelligent Robot Swarm for Attendance, Recognition, Cleaning, and Delivery.



Criterion 1 : Functionality

Evaluating individual robots to make sure that each robot can perform operations as expected.

Length	180mm
Width	100mm
Height	115mm
Wheel Diameter	60mm
Surface Contact/ wheel	$2mm^2$
Torque	(6v): 1.5 kg/cm / 20.86 oz./in
Sweep angle	180deg
Time/instruction loop	2 .7 seconds

Table 5. Functionality

Criterion 2- Robustness (robust behaviour):

This criterion is defined to test the ability of the robots to perform allocated tasks as much efficiently, effectively and robust as possible.

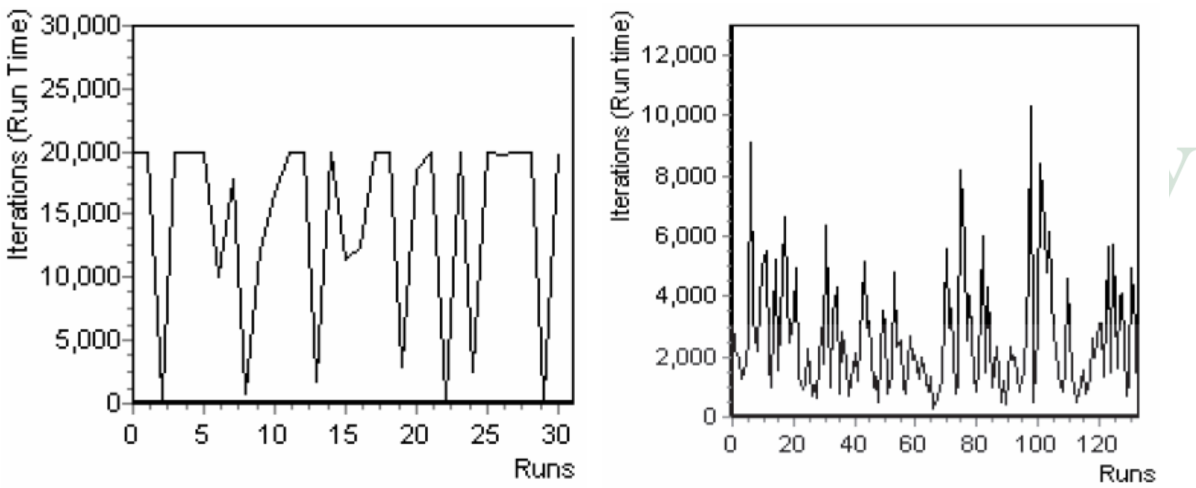


Figure 24. Results of time duration to complete observation tasks with (Left) static obstacles and goals (Right) Static Obstacles and movable goals

Criterion 3- Communication skill and information exchange:

This criterion is defined to test the ability to transmit and share correct information required by the staff as well as other robots at the correct times.

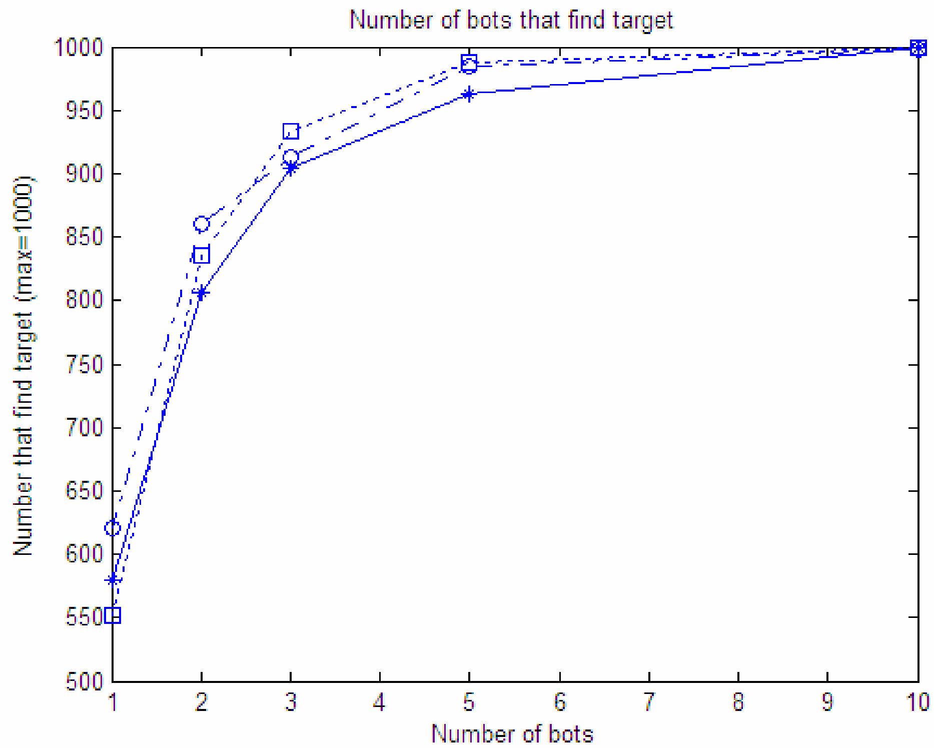


Figure 25. Number of successful searches (out of 1000) for differing number of bots. Results for max bot velocity of 0.4 m/s (squares), 0.65 m/s (circles), and 0.90 m/s (stars).

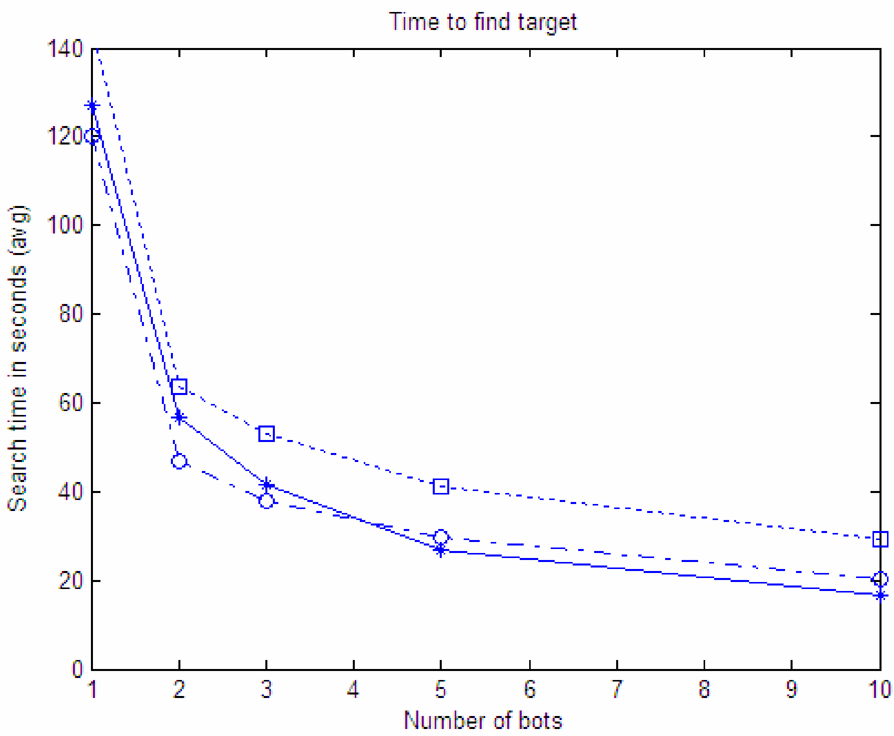


Figure 26. Comparison of search times for differing velocities and differing number of bots. Results for max bot velocity of 0.4 m/s (squares), 0.65 m/s (circles), and 0.90 m/s (stars).

Criterion 4- Timeliness and responsiveness:

This criterion is defined to test the ability to perform the tasks on time. The following example can be a good indicator for robot performance.

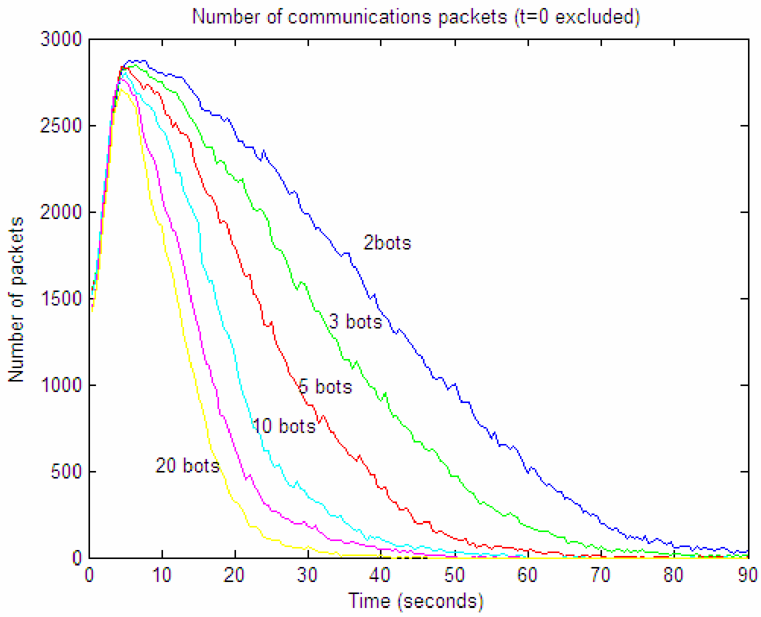


Figure 27. Comparison of number of packets transmitted versus time. The lines show the results for different numbers of bots in the search.

Criteria 5- Reliability:

This criterion is defined to test the ability to perform reliable actions.

The robots have found the food source 8 times for every 10 runs under 15 mins of the previous theoretical calculations and 10 times out of 10 for 25 mins for practical values.

The formation control does not have any standard way of measuring the response functions or feedback however the experimental results are already discussed in the previous section.

Technical Difficulties/Challenges

Ultrasonic Sensors

For this project, we used a commercially available HC SR04 Ultrasonic Sensor. Since these IR sensors are based on sound, the presence of sound absorbing materials in the natural environment give rise to noisy readings. Hence, we had to construct a special arena made of high density Styrofoam cardboard to eliminate noise do to sound absorbing material.

Additionally, the sound based Ultrasonic sensors are known to bounce of curved/smooth surface leading to a “Edge Effect”, in which the edges of a smooth object are not detected. Hence to prevent this, we built a square arena with squared object to prevent the bots from colliding along the edges

IR Communication

Developing an IR communication module in which we used 4 IR sensors, which worked in parallel was a challenge. There was no prior Arduino library available and we had to develop a C++ library for reading 80 bits tandem, coming from 4 different IR sensors (20 x 4 = 80bits) using a single TX/RX port in ATMEGA328P. Additionally, mapping 3 servo motors, 4 IR transmitters and 4 IR receivers was a challenge since Arduino Uno has only 14 pins in total.

3D Printing

3D printing the chassis and the hinges presented numerous mechanical challenges. Since we did not have any prior experience with 3D printing, we had to initially perform a lot of experiments with various materials and filling density. Additionally, the limited number of 3D printers in the I2P lab and a non-functional 3D printer in the library aggregated the scarcity of the printers. The chassis designed for this experiment was 17 cm x 8 cm. Since most commercially available 3D printers can only print a maximum dimension of 15cm x 15cm, we had to resize the original chassis design and eventually, resorted to using a commercially available aluminum chassis.

Formation Control

The lack of real time memory/data storage prohibited the use of the microcontroller and performance of the ant colony. Since most decisions in the biological world are based on past memory, the lack of memory of past events lead to real time memoryless swarm which could not truly perform/replicate the activity of true swarm. Since ATMEGA328P is not capable of executing parallel process, sampling of data and movement of the bot could not be performed parallelly.

Hardware Calibration

Procurement and calibration of hardware was an expensive and tedious process. Most of the associated hardware for the project was expensive, with long shipping times. The use of DC servo motors for the two wheels lead to serious calibration challenges, since the slightest inaccuracies in windings cause the motors to spin at different speeds (rpms) for a constant current. These inaccuracies lead to the bot with a constant side drag which had to be corrected for each experiment. Additionally, the friction between the wheels and the ground had to be accounted for different surface and the wheels speed had to controlled according to the surface.

References

[1] “Ant,” *Wikipedia*, 14-Dec-2017. [Online]. Available: https://en.wikipedia.org/wiki/Ant. [Accessed: 15-Dec-2017].

[2] J. Surowiecki, *The wisdom of crowds why the many are smarter than the few*. London: Abacus, 2014.

[3] S. Zielinski, “*Ants' hive mind*,” Science News, 23-Sep-2013. [Online]. Available: https://www.sciencenews.org/article/ants-hive-mind. [Accessed: 15-Dec-2017].

[4] A. Khan, “*Ant colonies: Powerful search engines that benefit from experience*,” Los Angeles Times, 28-May-2014. [Online]. Available: http://beta.latimes.com/science/sciencenow/la-sci-sn-foraging-ants-search-engine-google-bioinspired-20140527-story.html. [Accessed: 15-Dec-2017].

[5] Carrol CR, Janzen DH "*Ecology of foraging by ants*". Annual Review of Ecology and Systematics. 4: 231–257. doi:10.1146/annurev.es.04.110173.001311. 1973

[6] Banks, Alec; Vincent, Jonathan; Anyakoha, Chukwudi (July 2007). "*A review of particle swarm optimization. Part I: background and development*". Natural Computing. doi:10.1007/s11047-007-9049-5.

[7] Faigl, J.; Krajnik, T.; Chudoba, J.; Preucil, L.; Saska, M. *Low-Cost Embedded System for Relative Localization in Robotic Swarms*. In ICRA2013: Proceedings of 2013 IEEE International Conference on Robotics and Automation. 2013

[8] “*Research Advance in Swarm Robotics*,” Defence Technology, 31-Oct-2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221491471300024X#tbl1. [Accessed: 15-Dec-2017].

[9] “Multi-Robot Systems Lab - Rice University, Houston TX,” *Multi-Robot Systems Lab - Rice University, Houston TX*. [Online]. Available: http://mrsl.rice.edu/. [Accessed: 15-Dec-2017].

[10] “Jasmine,” *Swarmrobot* | Open-source micro-robotic project. [Online]. Available: http://www.swarmrobot.org/. [Accessed: 15-Dec-2017].

[11] “TERMES,” *Wyss Institute*, 21-Aug-2017. [Online]. Available: https://wyss.harvard.edu/media-post/termes/. [Accessed: 15-Dec-2017].

[12] Micropik, “MICROPIK,” *MICROPIK*. [Online]. Available: http://www.micropik.com/. [Accessed: 15-Dec-2017].

[13] Vishay, “Products,” Vishay. [Online]. Available: https://www.vishay.com/. [Accessed: 15-Dec-2017].

[14] 亞格網頁設計公司, “Slvia jiapo,” 億光電子工業股份有限公司. [Online]. Available: http://www.everlight.com/. [Accessed: 15-Dec-2017].

[15] Atmel Corporation - Microcontrollers, 32-bit, and touch solutions," Atmel Corporation - Microcontrollers, 32-bit, and touch solutions. [Online]. Available: http://www.atmel.com/. [Accessed: 15-Dec-2017].

[16] Jessica M.C. Pearce-Duvet Coen P.H. Elemans Donald H. Feener, Jr- Oxford Economic: Walking the line: search behavior and foraging success in ant species, Behavioral Ecology, Volume 22, Issue 3, 1 May 2011, Pages 501–509, https://doi.org/10.1093/beheco/arr001

[17] James McLurkin, S.B., M.I.T., The Ants: A Community of Microrobots, 1995.

[18] James McLurkin. Measuring the Accuracy of Distributed Algorithms on Multi-Robot Systems with Dynamic Network Topologies. Proceedings of the The 9th International Symposium on Distributed Autonomous Robotic Systems, November 2008.

[19] M. Schwager, J. McLurkin, J. J. E. Slotine, D. Rus. From Theory to Practice: Distributed Coverage Control Experiments with Groups of Robots. Proceedings of International Symposium on Experimental Robotics, July, 2008.

[20] Jonathan Bachrach, James McLurkin, and Anthony Grue. Proto Swarm: a language for programming multi-robot systems using the amorphous medium abstraction. Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent systems, 2008.

[21] Mac Schwager, James McLurkin, and Daniela Rus. "Distributed Coverage Control with Sensory Feedback for Networked Robots. Proceedings of Robotics: Science and Systems, August 2006.

[22] James McLurkin and Daniel Yamins. Dynamic Task Assignment in Robot Swarms. Proceedings of Robotics: Science and Systems, June 8, 2005.

[23] James McLurkin and Jennifer Smith. Distributed Algorithms for Dispersion in Indoor Environments using a Swarm of Autonomous Mobile Robots. Proceedings of Distributed Autonomous Robotic Systems, June 23, 2004.

[24] Anita M. Flynn, K.R. Udayakumar, David S. Barrett, James D. McLurkin, Dean L. Franck and Arthur N. Schectman. Tomorrow's Surgery: Micromotors and Microrobots for Minimally Invasive Procedures. Journal of Minimally Invasive Therapy and Allied Technologies, 1998:7/4:343-352.