Pseudo Swarm Implementation Using Robotic Agents and Graphical Simulation

Jeremiah C. Aboganda, Christopher Kim C. Go, Jonathan R. Maniago, Michelle Jean R. Sia, and Carlos M. Oppus

Ateneo de Manila University, Bachelor of Science in Computer Engineering
{jeremiahaboganda,gyu_lon,katipunan_kaidashi_kikou}@yahoo.com, michellejeansia@gmail.com, coppus@ateneo.edu

Abstract. The Pseudo Swarm Implementation aims to demonstrate the concept of swarm intelligence through the use of simulation and actual robotic units in hopes of recreating the complexity of natural systems. The actual swarm robots are modular in design and powered by PIC16f84 microprocessors. Each robot is composed of the motor module and a detachable sensor module. The motor module was constructed first and designed to handle basic movement and the random walk function. The sensor module was then designed afterwards which is capable of interfacing with the motor module through a universal protocol. The simulation platform was accomplished through the use of ActionScript3 to graphically depict the swarm behaviors given certain rules and a controlled environment. The simulation's purpose is to complement the data by creating robot objects similar to the actual robots and reproducing a larger number of them in a given environment. This work is a comprehensive study on a variation of the of swarm intelligence implementation of how a complex system may be composed of simple units defined by basic state machines.

Keywords: Swarm Behavior, Swarm Robotics, PIC16f84 Microcontroller, ActionScript 3.0, State Machines.

1 Introduction

Despite all of these advances in robotics or technologies in general, the use of increasingly sophisticated technologies have proved to be impractical due to increased costs and risks. Having reached this apparent limit in technological advancements, it is time to return to the fundamentals and natural designs of our world that have withstood both time and man. These natural designs or systems are what can be referred to as collective or global consciousness apparent in most basic life forms such as ants, termites and even microorganisms. Utilizing basic entities and intelligence, a cooperative network or structure, with the potential to accomplish communal goals or tasks, is created through the interactions. Perhaps it is possible to adapt these primal structures into the development of technologies today.

The aim of the paper is to simulate Pseudo-swarm Intelligence. The term "Pseudo" is coined due to the nature of the robotic agents to be developed. With the aim to develop low cost actual and simulated robotic agents and to remain faithful to the

S.S. Ge et al. (Eds.): ICSR 2010, LNAI 6414, pp. 11–19, 2010.

interaction of natural systems, direct wireless communication between robotic entities or agents is non-existent. Similar to the behavior of most agents of natural systems, these robotic agents rely on sensory data to be able to communicate and interact or react with stimuli from their environment.

The robotic agents are driven by PIC microcontrollers and are composed of three general modules: processor module, motor module and the sensory module. The sensory modules would consist of proximity and light sensing capabilities. With the sensory module being replaceable and the microcontroller being reprogrammable, robotic agent can be modified to be able to accomplish certain tasks or to demonstrate certain state machines.

The graphical Pseudo-swarm environment simulation is powered by Action Script 3.0. The simulation program is capable of producing a specified number swarm of agents to implement the same state machines governing the behaviors of the robotic agents. This part of the implementation addresses the limitations presented by the hardware in terms of number, consistency and apparent costs.

2 System Configuration

2.1 Robotic Agent

The swarm robot is composed of two primary modules. The primary modules were the Control or PIC Module and the Motor Module. Several sensor modules may be used to expand the functions and general capabilities of the swarm robot.

The swarm robots embody finite state machines of a minimal number of states. The common component state in most state machines was the “idle” behavior which was either the random walk routine or a no operation routine. As purely responsive agents, excitation of the states through the sensors expansion modules served as triggers to state shifting of the swarm robot.

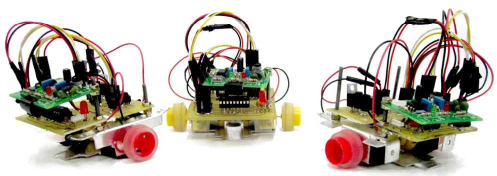

Fig. 1. Pseudo Swarm Bot Proximity Configuration

2.1.1 Control Module / PIC Module

The Control Module / PIC Module was mainly composed of the detachable PIC16F84A microcontroller. The PIC Module was designed to be similar to microcontroller trainer boards without onboard means of reprogramming the installed microprocessor. Reprogramming the installed microprocessor required the extraction of the microprocessor from the chip mount.

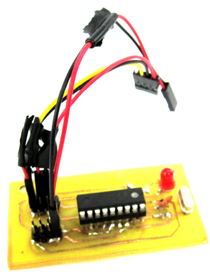

Fig. 2. Pseudo Swarm Bot Control Module

2.1.2 Motor Module

The Motor Module was mainly composed of a detachable L293D Motor Driver. The main function of the Motor Module was to serve as both the motor control and the power regulator of the entire swarm bot.

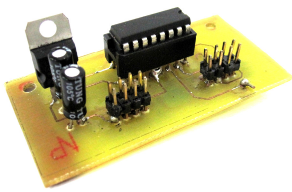

Fig. 3. Pseudo Swarm Bot Motor Module

With the available power lines and numerous configurable input pins from the PIC module, additional modules can easily be incorporated into the swarm robot. Examples of such sensor module expansions would be the proximity sensor and the LDR light sensor.

2.2 Simulated Pseudo Swarm Environment

For the initial tests, Action Script 3.0 is used to implement some pseudo-swarm algorithms. These algorithms are conceived and based from the limited functionality of the actual robotic agents. Pure frame-based events are used in the simulation.

2.2.1 Simulated Pseudo-Swarm Robot Design

The simulated swarm robot is composed of `two objects: the Sensor object and the Body object. Combined, these two objects simulate a pseudo-swarm robot.

The swarm robot "sees" something whenever an object overlaps with the Sensor object. Likewise, it is seen whenever a Sensor of another robot comes across with the body object. There would be more complicated detections in the upcoming sections, but this basic concept is used all throughout.

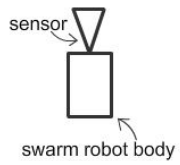

Fig. 4. Simulated robot composition

The concept of rotating the swarm robot is simple. The Body and Sensor objects are placed within the robot such that the center of the Body object is the center of rotation. Whether the robot would turn clockwise or counter-clockwise is set randomly at the start by the environment.

2.2.2 The Environment Design

The simulated robot agents are put inside an environment wherein they would interact with other swarm robots and the environment itself.

The environment is composed of a bounding box and the robot agents themselves. The robot agents, through their Sensor objects, are able to detect the Boundaries and are able to react to it.

Introduction of elements like Light and Food will be added later on to simulate more complicated algorithms.

The environment also incorporates rules like the speed and the rotational speed of the simulated robot agents. It also determines the number of robots inside the arena. It also makes sure that the initial positions of the simulated robots are distinct, such that the robots do not overlap one another

2.2.3 Pure Random Walk with Clumping

The first algorithm simulated by the group is the pure random walk algorithm. It consists of robots moving randomly inside a closed fixed space defined by the boundaries set in the environment.

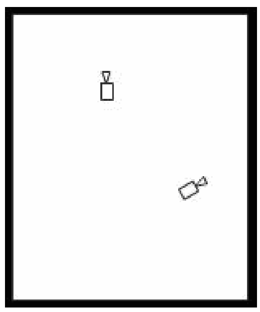

Fig. 5. Environment setup concept

Whenever these robots detects one another, it stops and waits for a random time, then checks if there is no obstacle. If a robot is in the way, it waits again for another random time. If there is no robot in the way, it continues moving forward. Eventually, two simulated robots would have their Sensor object overlap with the body object of the other, thus refreshing the wait time indefinitely. They would never move from their positions while the simulation is running. These two robots would form the beginning of a clump. This implies that if a third robot would happen to pass by and detect any of the first two non-moving robots, it would wait too for an indefinite amount of time, and so on.

Table 1. Agent State Machine for Clumping

State	Action
0	Robot moves forward, stop while waitTime > 0
1	Robot detected a Boundary object, do rotate action

2.2.4 Pure Random Walk with No Clumping

The group has also created a version of the random walk algorithm with no clumping. Majority is the same with the earlier algorithm, however with this algorithm, while the robot cycles through its waiting time, it moves backward at a set speed provided by the environment variable.

Table 2. Agent State Machine for Non-Clumping

State	Action
0	Robot moves forward, move backward while waitTime > 0
1	Robot detected a Boundary object, do rotate action

2.2.5 Light Detection, Clumping at the End of the Light

This algorithm introduces the Light object. Having the Light object simply requires an extra robot state in the simulated robots, in addition to their default random walk process.

This algorithm does not use the Sensor object of the robot because the actual built robotic agents have the option to attach additional light sensors. Therefore, the trigger must be when the Light object makes contact with the body of the simulated robotic agent. Once this occurs, the robotic agent will cycle through two states to maintain its position within the lighted area. While within the lighted area, the agent will continue moving forward. However, upon leaving the area, the agent will then retrace its steps until it finds the light trigger once again.

Table 3. Agent State Machine for Light Detection

State	Action
0	Robot moves forward, move backward while waitTime > 0
1	Robot detected a Boundary object, do rotate action
2	Move the robot backward, placing it in the light.

2.2.6 Line Attempt Algorithm

This algorithm attempts to make the simulated swarm robots form a line out of the detection and communication limitations of the pseudo-swarm robots' set-up. In this attempted algorithm, the robot agents react to a "Food" trigger. Once an agent detects the trigger, the agent halts all activities and produces its own area with the same characteristics as the initial trigger. The end result would then be extending trigger area and an elongated version of the results of the clumping algorithm.

Table 4. Agent State Machine for Line Attempt

State	Action
0	Robot moves forward, move backward while waitTime > 0
1	Robot detected a Boundary object, do rotate action
2	Stop robot in the tracks, generate a SensorField around the object

3 Results and Discussion

Four robotic agents were constructed and tested for random walk, random walk with light seeking and random walk with proximity sense trials.

3.1 Robotic Agents

The random walk routine, whose permutations were predominantly forward, produced different results per swarm robotic unit. Despite having the similar high and low seeds in the programs of the four swarm robots, each one behaved differently. Each unit demonstrated varying bursts of displacement per movement routine. These tests are usually terminated once the agents are rendered immobile by corners or other robotic agents.

The units were then fitted with proximity sensors. This enhanced the test length to last the entire run of the battery life as units were able to avoid both the boundaries and other obstacles while displaying the same random burst movements.

Light seek and clumping was also on configuration which was tested. With the custom built light sensors, the robotic agents were able to find the lighted portion of the field. Success rates, however, were reduced due to limited battery life.

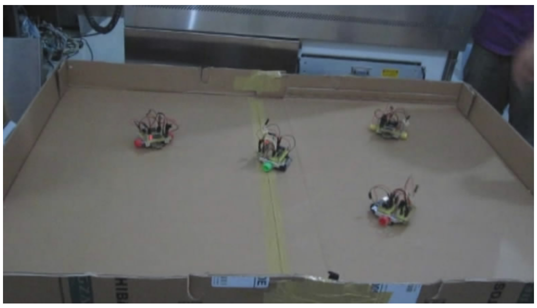

Fig. 6. Simulation Trial Demonstrating the Random Walk Routine

Fig. 7. Simulation Trial Demonstrating the Light Seek Routine

3.2 Simulated Robotic Agents

The simulated environment was able recreate the conditions of the robotic agents trials. The simulations, however, were limited by the hardware the platform was running on. Due to the incapability of the programming language used to pipeline functions and allocate proper memory units, the simulation environment, run in a relatively high end laptop, can simulate up to 50 agents before experiencing lag. The more complex algorithms demonstrated an even heavier toll on the computer hardware.

Despite the platform handicaps, the simulated environment and agents, particularly the random walk with clumping algorithm, was rerun and confirmed to produce consistently random outcomes.

The other algorithms also produced uniformly random outcomes. The Light Detection and Clumping Algorithm were effectively able to collect the agents at the light boundaries to form a makeshift wall.

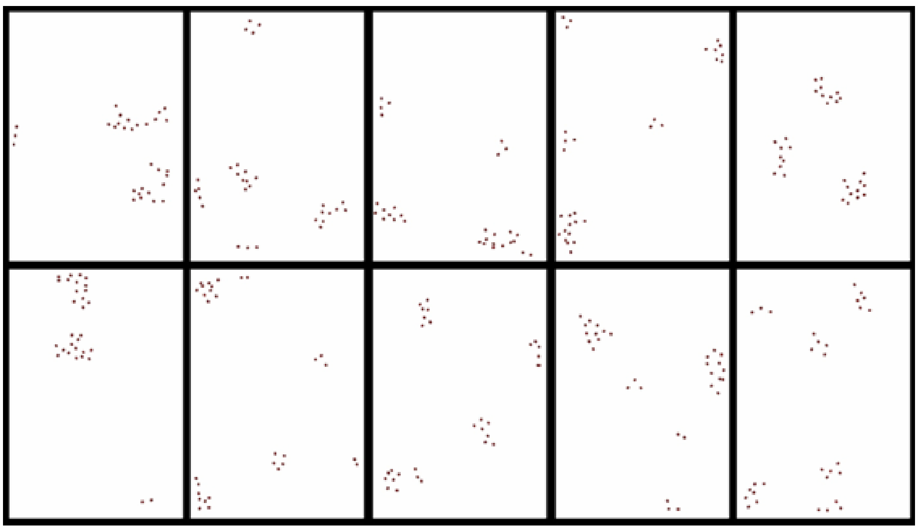

Fig. 8. Simulation for Random Walk and Clumping Algorithm with 30 agents

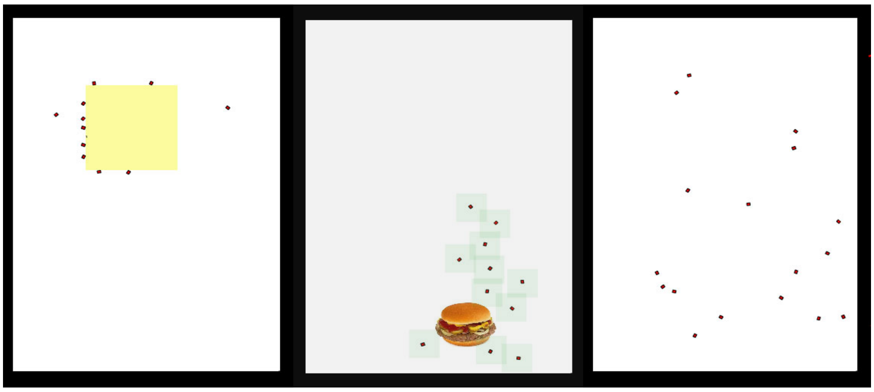

Fig. 9. Simulation Platform demonstrating Light Detect, Line Attempt and Random Walk

The initial random walk state, found in all algorithms, served to be a relatively effective search mechanism for the robotic agents. By increasing the number of robotic agents, the agility of the swarm to locate a particular trigger is also increased.

The Line Attempt Algorithm as well was able to form elongated clumps mimicking that of a line. This, however, would vary given a much larger environment or a differently sized food trigger object.

4 Conclusion and Recommendations

The natural order is almost always composed of numerous agents enclosed within a given an environment. Often times, these natural orders or systems demonstrate a consistent degree of uniformity amidst complexity that our own artificial systems are incapable of replicating.

The study of swarm or pseudo swarm intelligence gives new avenues of understanding these natural systems and how these natural systems can be used in different fields. Swarm intelligence or the concepts thereof have a place in both software and hardware designs.

For robotics, swarm emphasizes the use of basic units thus lowering the cost and increasing the robustness of a fleet or system of autonomous machines. The dispersal and clumping capabilities of swarm robotic agents can be applied physically to other tasks such as fire seeking, oil spill cleaning and search and rescue. For the medical and biological fields, swarm intelligence may be observed in the behavior and characteristic of most simple cellular organisms. From the bacterial level to the level of most colony life forms as termites and ants, swarm intelligence provides a means of completing tasks or goals with the use of simple rules and conditions from which an organizational structure composed of far more numerous states and possibilities may emerge.

References

1. Engelbrecht, A.P.: Fundamentals of computational swarm intelligence. Wiley, Hoboken (2007) (c2005)
2. Kennedy, J.F.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (c2001)

3. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems
4. Russel, S., Norvig, P.: Artificial Intelligence, A Modern Approach
5. Bonabeau, E.: Swarm intelligence: from natural to artificial systems. Oxford University Press, New York (1999)
6. Murphy, R.: Introduction to AI robotics. MIT Press, Cambridge (c2000)