SWARM INTELLIGENCE WITH LEGO NXT

B Samanta
Department of Mechanical &Electrical Engineering
Georgia Southern University, GA

Presentation Overview

- Introduction
- LEGO NXT Robot platform
- PSO Algorithms
- Implementation Results
- Conclusions
- Future work

Introduction

- Robotics is viewed as an emerging field that has potential to significantly impact
  - the nature of engineering and science education at all levels, from K-12 to graduate school
- A recent development in robotics is swarm robotics
- The use of a large group (swarm)
  - of small, simple and cheaper robots with limited local processing capability
  - in place of a large, powerful and expensive robot is being envisioned in many hazardous, unknown and dynamic environments.

Swarm Robotics

- The advantages of using swarms instead of a single centralized robot include enhanced capabilities in terms of
  - wider dynamic coverage,
  - fault tolerance,
  - self organization and
  - emergence.
- Application areas of robot swarms include:
  - autonomous search and rescue operation,
  - decentralized autonomous systems for protection and damage control, among others.
- For successful implementation, both hardware and software issues of such co-operative robots need proper investigations

The Present Work

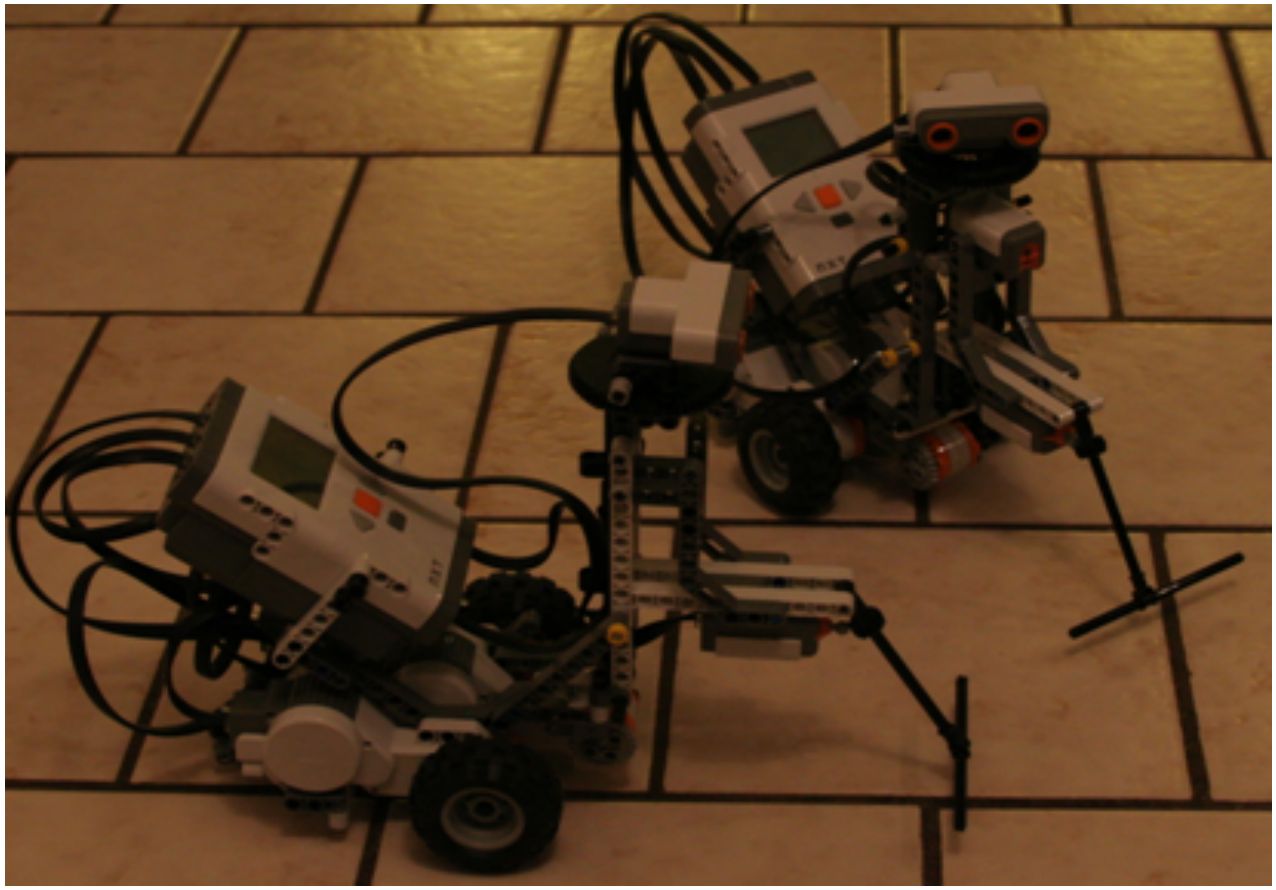
- An exploratory (pilot) project on swarm robotics was initiated in Summer with a team of
  - two high school students
  - and a senior Mechanical Engineering undergraduate

- The plan was to test the feasibility with a small group.
- A group of three simple mobile robots (Lego NXT) was used to study 'search and rescue' operation in the context of swarm robotics.
- A bio-inspired global optimization technique called particle swarm optimization (PSO) was used as the main algorithm.

LEGO NXT Platform

- Hardware
- Lego Mindstorms NXT robotics kit was used as the robotic platform for the project. Four kits were procured, at any time three were used and the fourth was kept as a stand-by.
- The main component of a NXT kit is the NXT intelligent brick.
  - Two microcontrollers embedded inside the brick.
  - One brick can take inputs from four sensors and control up to three motors at once with four input ports for sensors and three output ports for motors.
  - The NXT kit comes with four sensors, namely, light, sound, touch and ultrasonic sensors.
  - In this project only two of these sensors, touch and ultrasonic sensors were used.
  - Two of the motors were used for driving the robot.
  - The third motor was used to rotate the ultrasonic sensor through a geared mechanism.
  - Figure 1 shows the photograph of two assembled NXT mobile robots. For communication with a PC laptop, the wireless Bluetooth connection was used.
  - There is a speaker for sound and a LCD display for the status of NXT.

LEGO NXT Platform



LEGO NXT Platform

- NXT Brick
  - Atmel 32-bit ARM7 processor running at 48 MHz with 64 KB of RAM and 256 KB bytes flash memory.
  - An Atmel 8-bit AVR running at 8 MHz with 512 KB RAM and 4KB of flash memory.
  - The AVR controls the peripherals while ARM7 has the main processing power.
  - The 100 x 64 LCD display is used to navigate through the NXT menu and shows the status of the brick.
- *Servo Motors*
  - Each motor has a built-in tachometer to keep track of the motor rotation with an accuracy of ±1°.
  - Each motor is also equipped with a servo loop for velocity and position control.
- *Touch Sensor*
  - The touch sensor senses when it is pressed and when it is released. This signals to the robot that it has contacted another object.
  - The sensor was activated using a link in front of it.
  - When the robot hits an object with the link activating the touch sensor, the NXT was programmed to stop moving forward.

LEGO NXT Platform

- *Ultrasonic Sensor*
  - The ultrasonic sensor was used for the robot to avoid obstacles and measure distance.
  - The rotating platform was used programmed so that the ultrasonic sensor can cover the range of -90° to 90° in front of the robot.
  - The sensor measures distance by calculating the time it takes for a sound wave to hit an object and return. It measures a distance upto 255 cm with a precision of ±3 cm.
- *Bluetooth*
  - The communication between any NXT robot and the PC laptop (host) was implemented using a D-Link DBT-120, wireless Bluetooth 2.0 USB Adapter.
  - It is compatible with Windows 2000/XP, follows the IEEE 802.15.1 standard, uses USB 2.0 interface, and sends signals at 2.1Mb/s. The Bluetooth USB Adapter supports the Microsoft Service Pack 2 Bluetooth stack.

LEGO NXT Platform

- *Java Software Platform for NXT*
  - The NXT needs to have a firmware installed in order to be usable.
  - The default firmware and software, NXT-G, are adequate for normal users.
  - However, for greater flexibility, an alternate firmware and software system for the NXT, called leJOS NXJ, was adopted for the project. It interfaces with the NXT hardware and allows users to program in Java.
  - The PC laptop was using leJOS JVM (Java Virtual Machine) under Linux operating system.
  - An open source integrated development environment (IDE) suitable for leJOS NXJ, called Eclipse, was used in this project.

PSO

- Particle swam optimization (PSO) was proposed by Kennedy and Eberhart
  - a population based stochastic optimization technique
  - inspired by the social behavior of bird flocking.
- PSO is an algorithm based on the group (swarm) behavior.
- The algorithm searches for the optimal value by sharing
  - the cognitive and
  - the social information

  among the individuals (particles) in the global solution space.
- PSO has many advantages over other evolutionary computation techniques like genetic algorithms (GA) such as
  - simpler implementation,
  - faster convergence rate and
  - fewer parameters to adjust.
- The popularity of PSO is growing with applications in diverse fields of engineering, biomedical and social sciences, among others.

PSO Algorithms

- PSO Model

$$\text{velocity}: v_{ij}(k+1) = v_{ij}(k) + c_1 U(0,1)[p_{bij}(k) - x_{ij}(k)] + c_2 U(0,1)[p_{b\,gj}(k) - x_{ij}(k)],$$

$$\text{position}: x_{ij}(k+1) = x_{ij}(k) + v_{ij}(k+1), \qquad i = 1, m, \;\; j = 1, n, k = 1, N$$

  - $c_1=c_2=2$; $r_1=r_2=U(0,1)$ uniform random numbers
- DPSO Model
- Each robot would measure its position, update its velocity and position,
- update its personal best value (*pbest*) and its personal best position.
- Each robot would send to the PC the values of its current position, and its personal best.
- The PC will receive the *pbest* values of all robots and transmit to them the global best (*gbest*) to be used by each robot for updating its velocity and position for the next move.

DPSO Based Implementation

- The aim is to select the robot positions (x_l, y_l) to minimize the objective function:

$$J = \sum_{l=1}^{N}\left((x_d - x_l)^2 + (y_d - y_l)^2\right)$$

  - where *l* is the robot index,
  - N represents the total number of robots,
  - (x_d, y_d) is the desired position and
  - (x_l, y_l) represents the current position of the robot '*l*'.

- DPSO was used from a user-given range for each robot position [-400, 400] mm to cover the entire search space.

Project Objectives

- The project objectives were to
  - get familiarized with the Lego Mindstorms NXT hardware,
  - interface the sensors and actuators (motors),
  - program the NXT brick using leJOS NXJ,
  - establish communication between NXT robots and a PC laptop using Bluetooth,
  - program each NXT for uniform spiral coverage using Archimedes' curve, and
  - implement the distributed particle swarm optimization (DPSO) algorithm for 'search and rescue' operation using a swarm of NXT mobile robots.

Implementation Results

- The coverage and target reaching was implemented in a decoupled manner.
- The uniform spiral coverage using Archimedes' curve was implemented in leJOS NXJ.
- The pseudocode for DPSOwas implemented in leJOS NXJ.
- The codes were developed in Eclipse IDE and the firmware was uploaded to each NXT.
- The NXT robots were placed in three known positions in the lab for this project.
- In the first phase, the robots would search for an object in covering the respective local area in spirals.

Implementation Results

- The coverage and target reaching were implemented in a decoupled manner.
- The uniform spiral coverage using Archimedes' curve was implemented in leJOS NXJ.
- The pseudocode for DPSOwas implemented in leJOS NXJ.
- The codes were developed in Eclipse IDE and the firmware was uploaded to each NXT.
- The NXT robots were placed in three known positions in the lab for this project.
- In the first phase, the robots would search for an object in covering the respective local area in spirals.

Table 1: The pseudo code for distributed PSO (DPSO) algorithm

```
Initialize the swarm
While (mission on)
      get sensor readings
  If (no object is found) Then
       move in spiral
   Else
     evaluate the target position
     report the target position to the host
     set dPSO on
          While (dPSO on)
            get sensor readings
            update local best
            update global best;
                  report to host the current position and
            get global best
            move to the next best position
            avoid obstacle(s), if present
                                 If (target reached)  set dPSO off
          End /end while (dPSO off)
       Endif /endif/
   If (mission completed) set mission off
End /end while/completion of mission/
```

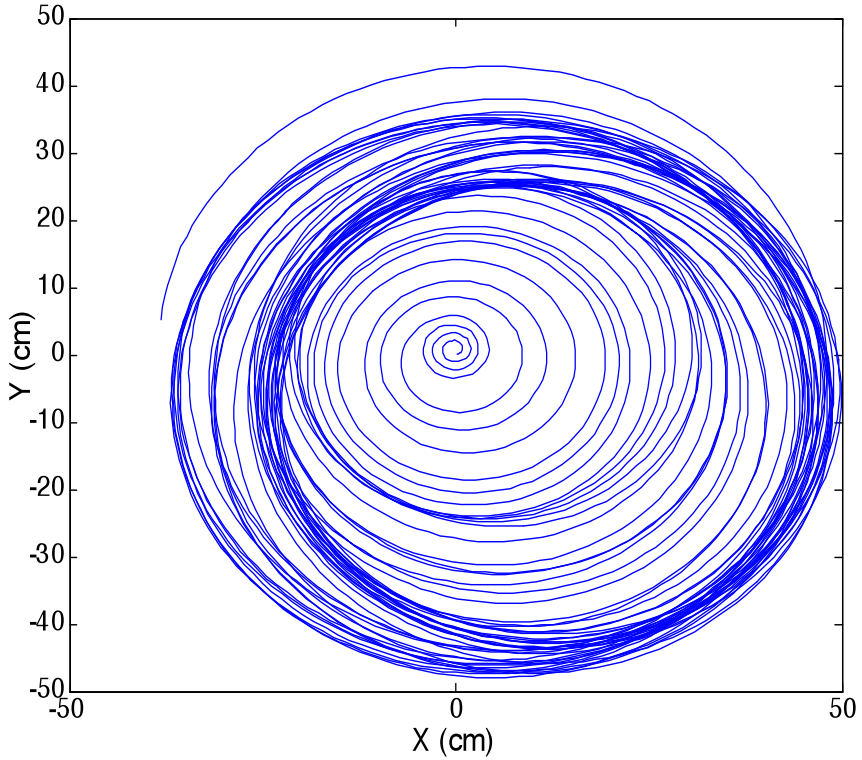


Fig. 7. Area coverage (spiral)

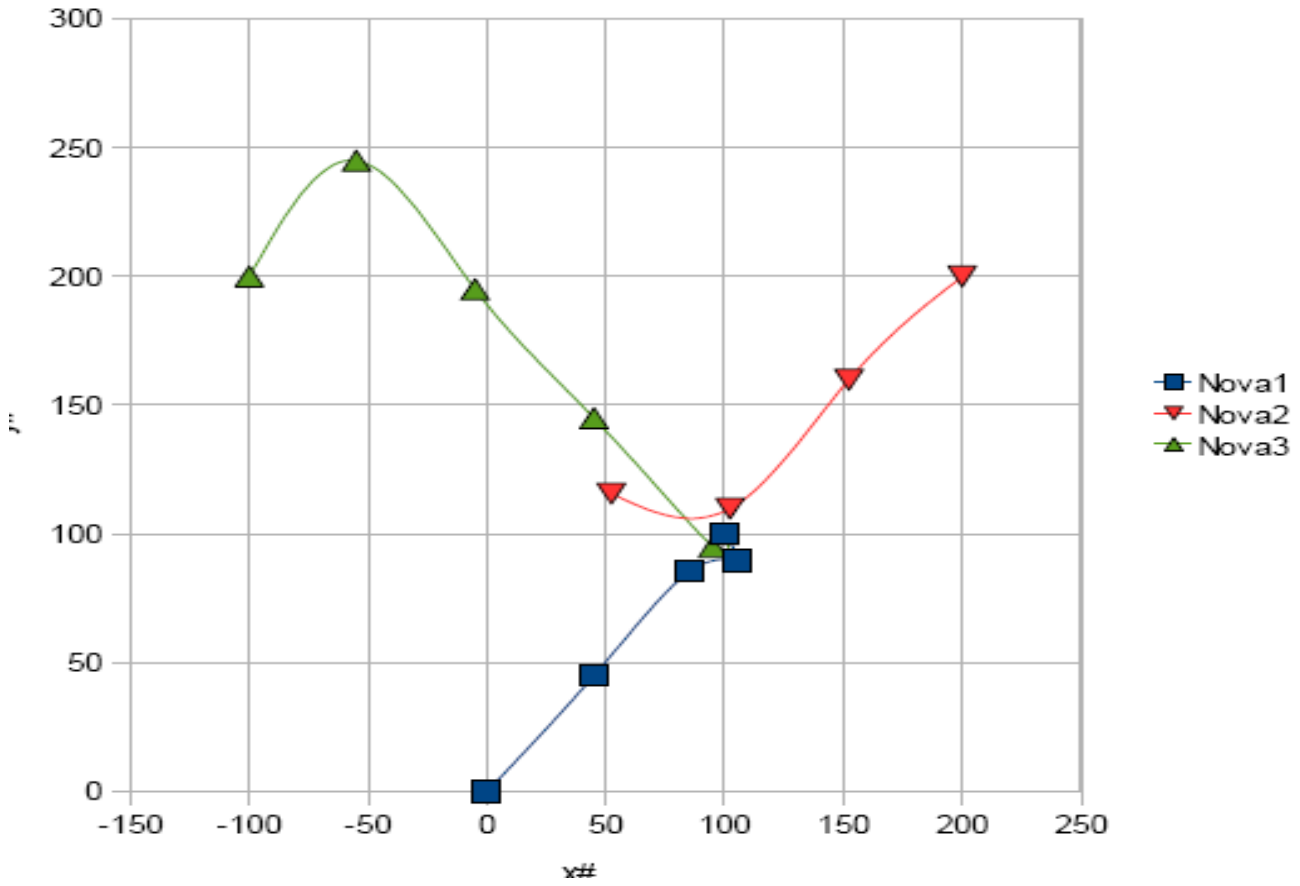


Fig. 8. Paths of robot swarm

Conclusions

- We present an exploratory project on swarm robotics involving high school students.
- The project provides educational and research experiences to the students covering a wide range of areas like sensing and actuation, control, swarm intelligence, hardware and software implementation.
- The students got a first-hand experience of implementing software codes in real-time engineering applications.
- However, some hardware and software issues would need closer and more thorough investigations for further enhancements.
- These are currently under consideration for the next phase.
- For example, the seamless implementation of coverage, search and rescue would be considered using a multimodal approach where the operation would switch from one mode (coverage) to the other (search and rescue).
- The involvement of larger groups of students with potential external funding is also envisioned in the future.

- THANKS