computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

UB SWARM: HARDWARE IMPLEMENTATION OF HETEROGENEOUS SWARM ROBOT WITH FAULT DETECTION AND POWER MANAGEMENT

Madhav Patil, Tamer Abukhalil, Sarosh Patel, Tarek Sobh

RISC Laboratory, University of Bridgeport, Bridgeport, USA, 06604
mpatil@my.bridgeport.edu, tabukh@my.bridgeport.edu, saroshp@bridgeport.edu, sobh@bridgeport.edu

Abstract: In this work we present the hardware architecture of a mobile heterogeneous robot swarm, designed and implemented at the Interdisciplinary Robotics, Intelligent Sensing and Control (RISC) Laboratory, University of Bridgeport. Most of the recent advances in swarm robotics have mainly focused on homogeneous robot swarms and their applications. Developing and coordinating a multi-agent robot system with heterogeneity and a larger behavioral repertoire is a great challenge. To give swarm hardware heterogeneity we have equipped each swarm robot with different set of sensors, actuators, control and communication units, power supply, and an interconnection mechanism. This paper discusses the hardware heterogeneity of the robotic swarm and its challenges. Another issue addressed in paper is the active power management of the robotic agents. The power consumption of each robot in the UB robot swarm is calculated and the power management technique is also explained in this paper. We applied this heterogeneous robot swarm to perform three sample tasks – Mapping task, human rescue task and wall painting task.

Keywords: Heterogeneous Swarm Robot, Hardware Implementation, Power Consumption in Swarm Robot, Hardware Design of UB swarm, Fault Detection in Heterogeneous Swarm.

1. INTRODUCTION

The Swarm robotics has been an emerging research paradigm over the last decade, inspired by group behavior animals including ants, bees, and other insects [1]. To date, most existing swarm robot systems have been designed and implemented with homogeneous hardware. Only a few of them included heterogeneous robots, but those swarm systems were limited physically and behaviorally. Due to the lack of methods and tools, swarm robot designers cannot achieve the complexity required for the real world applications [2]. The complexity of designing and physically implementing the heterogeneous robot swarm is greater when compared to the homogeneous robot swarms. There are several aspects involved in the development of robot swarm hardware, such as locomotion, actuation, navigation, size, appropriate sensors, cost, and communication [3]. One of the challenges for robot swarm is its autonomy, as the robot must be aware of its battery life, self-localization, etc. In our review article [3], we detail the hardware architecture of robot swarms with self-configurability, self-assembly, and self-replication. After reviewing existing swarm systems and studying the limitations, we decided to design and build our own robot swarm system. In this design we have considered some important factors such as the size, cost, autonomy, flexibility, robustness, power consumption, and weight of the robots. The main goal of our research is to build a heterogeneous robot swarm system in which each robot has distinct type of hardware compared to other robots. The proposed architecture is an autonomous, modular, heterogeneous robot swarm with self-configurability, self-assembly, and self-learning capabilities. Currently, electronic products are cheaper, smaller, lighter in weight and easily available, which makes robot swarms more cost efficient and compact in size [4].

The swarm-bot research project [5], deals with design and implementation of swarm robots (s-bots) with self-organizing and self-assembling capabilities, but each S-bot is physically identical (homogeneous) and uses the same kind of sensors, actuators, microcontrollers. S-bots can connect with other S-bots with a rigid gripper and are also able to lift the other S-bots to collaboratively create a bigger structure. Further, swarm-bots have been extended into a swarmanoid project, which is focused on the study, design and implementation of swarm systems with heterogeneous robots [6]. In this case, a swarm includes robots that can move on the ground, fly, and climb on vertical surface. In the swarmanoid project [6], robots use different colored light emitting diodes (LED) and omnidirectional camera for communicating with each other. The camera is pointed at a half spherical mirror to directly acquire images from its surroundings. The problem with swarm bot is that the images the camera receives are further away than seen in the mirror. Table 1 summarizes the hardware platforms implemented so far in swarm robot research experiments.

Table 1. Hardware Platform Summary.

Sr. No	Name	Sensor	Actuation	Controller	Communication	Positioning system
1	E Puck	11 IR, Contact ring, Color camera	wheeled	dsPIC	Bluetooth	Expansion IR based
2	Alice	IR, Light Sensor, Linear Camera	wheeled	Microchip PIC	Radio (115 kbit/s)	------
3	Jasmine	8 IR	wheeled	2 ATMega	IR	Integrated IR based
4	I-swarm	Solar cell	3 micro leg piezo-electric actuator	Not Available	Not Available	-----
5	Khepera	8 IR	wheeled	Motorola MC66831	RS232, Wired link	-----
6	Khepera III	11 IR, 5 Ultrasound	wheeled	PXA-255, Linux, dsPIC	WiFi& Bluetooth	Expansion IR based
7	S-Bot	15 Proximity, Omnidirectional Camera, Microphone, Temperature	wheeled, 2 gripper	Xscale Linux PICs	WiFi	Camera based
8	Swarm Bot	IR, Camera, Light, Contact	wheeled	ARM and FPGA 200 kgate	IR based	Integrated IR based
9	Kobot	8 IR, Color camera	wheeled	PXA-255, PICs	ZigBee	Integrated IR based

The hardware platforms described in the above Table 1 are homogeneous in nature and limited with capabilities and functionality. In Section 2 we explain the hardware architecture and the design goals of the UB robot swarm; Section 3 describes the sensory platform and their technical specification and working principles; Section 4 describes the locomotion and manipulation; Section 5 describes the communication and control units used on the UB robot swarm; and finally Section 6 shows an experimental results of human rescue task using the UB robot swarm.

2. HARDWARE DESIGN

The hardware design for any swarm is an interactive and an important phase; as all components and/or parts are assembled to build one robot swarm. At the hardware level, the most work has been done in collective behavior with homogeneous robots. In this project we decided to exploit reconfigurability and modularity using heterogeneous robots with decentralized control algorithms, which are influenced by the behaviors of ants, bee colonies, and insects in general [4]. Swarm robots developed so far are aimed to provide a research platform and not intended for real-world applications or vice versa [7]. In this section, we explain the hardware architecture of the UB robot swarm, design and built at the RISC Lab., University of Bridgeport. This swarm of heterogeneous robots is designed for real physical world applications in order to perceive their environmental physical properties through sensors and undertake manipulation and localization using actuators [7]. UB swarm robots can be used for real life applications as well as for research purposes. This modular hardware architecture consists of independent sensory units, actuator modules, and communication units, making the swarm system scalable and flexible such that more sensors and/or actuators can be added without modifying the overall architecture. Fig. 1 shows an overview of the hardware design implementation.

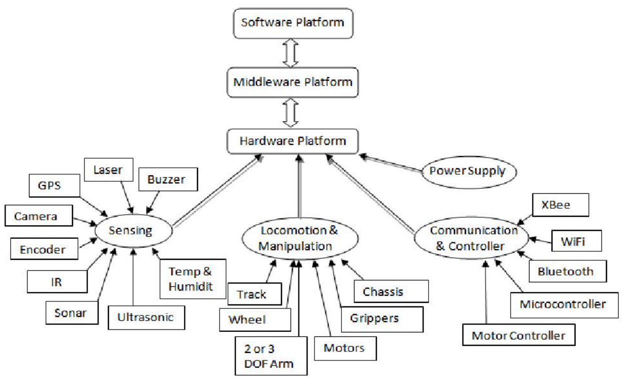

Fig. 1 – Hardware Architecture Design

There are many factors that have to be considered while designing and implementing the hardware platform for the heterogeneous robots. Following are the design goals for the UB swarm of heterogeneous robots, such as:

- Each robot should be easily modifiable and compatible with a high performance microcontroller.
- Should consume less power.
- Should provide user friendly mobile, modular, and flexible platforms.
- They should be reconfigurable and provide easy support for the software as well as for the middleware.
- They should provide low cost wireless communication for indoor as well as outdoor applications.
- They should have enough future expansion space for sensory units and actuators.
- The robot should be relatively of different size and shape with light weight, so that it can allow ease of movement and maneuverability.
- Each robot should be fully functional, and continuously coordinate and communicate with other robots.

Building of such a heterogeneous swarm of robots is a very complex task in real life. At the time of writing this paper we have built five swarm robots, all of which are fully assembled and tested for mapping, obstacle avoidance, painting, and rescue application [8]. The UB robot swarm is simple, capable of sensing, localization and actuation based on the local information and basic rules. In the following sections, the mechanical and electronic modules of the robots are described with their full capabilities. All the parts were tested and slightly modified for the applications, and then assembled to build the physical robot swarm. The software scans for replaced or extra added sensors itself which makes robot swarms more dynamic.

3. SENSORY PLATFORM

Gathering information or data about the working environment or surrounding environment of the swarm robots is an everlasting job. The sensory unit is important for robot swarms to perform tasks such as obstacle detection and avoidance, neighboring robot detection, and navigation [1]. Sensors are classified as five sensing elements of the robot swarm and are used to collect the information about their surrounding environment by means of electrical or electromechanical signals. In this proposed hardware design, each robot swarm is equipped with different types of sensors such as a temperature sensor, humidity sensors, an encoder, a camera, communication devices, proximity sensors, a ranger detector, and GPS tracking devices, etc. There are two primary factors that affect the limitation of sensors: the first is Range and resolution of the sensors, and the second is Noise that affects the output of the sensors.

The study of animal behavior shows that sensory skills are developed and adapted by the interpretation of signals generated from sensors [10]. In swarm robots, this self-learning capability is achieved by configuring and calibrating sensors for a given task [11]. Using multiple sensors [12] (known as sensor fusion) provides the most efficient and effective methods for collecting, and investigating the unknown environments. In this section explain all of the sensors that are used in our proposed robot swarm hardware with their respective technical specifications.

3.1 PROXIMITY SENSORS

Distance measurement and obstacle avoidance is the fundamental element of the information gathering quest. In swarm robotics, obstacle detection and collision avoidance in real time while the robots are in motion is major constraint and difficult task. Proximity sensors detect the object, surrounding material or other moving swarm robots without any physical contact, and calculate the very precise distance of that object [12]. This crucial component not only avoids collision, but also prevents the physical damage to the swarm robots and maintains safe distance [11]. Depending on the type of technology used, proximity sensors are classified into different categories such as inductive, capacitive, photoelectric, and ultrasonic proximity sensors.

Among these, ultrasonic proximity sensors were found to be more accurate and have more capabilities when compared to the others types of proximity sensors [7]. In proposed swarm robot model, we use ultrasonic as well as photoelectric (Infrared) proximity sensors.

3.1.1 ULTRASONIC SENSORS

Ultrasonic sensors are very commonly used to measure distance because they are inexpensive and easy to handle. They are used to avoid obstacles, to navigate, and for map building. Ultrasonic sensors emit sound waves (ultrasound) of 20 KHz frequency and use it to find a way around an obstacle, detect the uneven surfaces, any shape and size of object in known as well as in unknown environment. This is known as Echolocation. This sensor sends outs ultrasonic waves which are then detected after they are reflected or bounced back from object and/or obstacle. The time required for sending and to receiving the ultrasonic waves is measured and further processed to calculate the distance. These sensors are very precise in measurement and used in applications that require measurement between stationary and moving objects.

In our proposed hardware architecture design, ultrasonic sensors as shown in Fig. 2, are mounted on the sides (left and right), front and back corners of the robot. Following are the ultrasonic sensors used in UB robot swarm system with their technical specifications.

- Devantech SRF02 – We use the SRF02 in Serial mode, the mode pin is connected to 0v Ground. The Rx pin is data into the SRF02 and connected to the Tx pin on PIC controller. The Tx pin is data out of the SRF02 and connected to the Rx pin on PIC controller.
- Seeedstudio Ultrasonic Range Finder – This sensor operates on 5 VDC voltage, 15 mA current and the maximum measuring range is 400 cm. The data pin of sensor is connected to the digital pin of microcontroller.
- Ping Ultrasonic Sensor – The output from the ping sensor is a variable-width pulse that corresponds to the distance to target. The GND pin is connected to the GND of the microcontroller, 5 VDC is connected to the 5 VDC power supply and the signal pin is connected to the analog pin of the micro controller.
- LV-MaxSonar-EZ1 MB1010 Sensor - The analog pin of the sensor is connected to the analog pin of the controller. The analog voltage pin outputs a voltage which corresponds to the distance. The distance of an object from the sensor is directly proportional to the voltage.

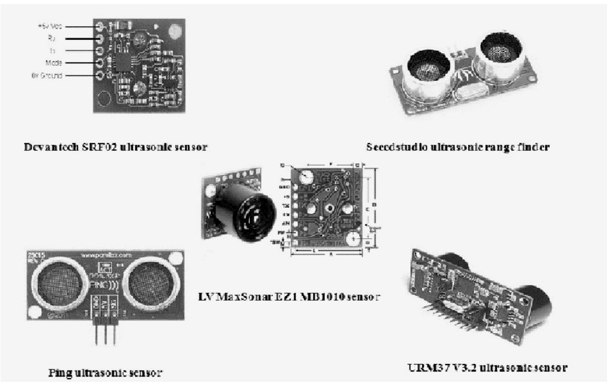

Fig. 2 – Ultrasonic Sensors used in UB Swarm

3.1.2 INFRARED SENSORS

The IR Range Finder works by the process of triangulation. A light pulse of wavelength range 850 nm (+/-70nm) is emitted from the sensor and then reflected back by an object or not reflected at all. When the light returns it comes back at an angle that is dependent on the distance of the reflecting object as shown in Fig. 3. Triangulation works by detecting this reflected beam angle and by knowing the angle, the distance can then be determined. The performance of the IR sensor is limited by its poor tolerance to the ambient light or bright object color reflection [13]. The IR range finder receiver has a special precision lens that transmits the reflected light onto an enclosed linear CCD array based on the triangulation angle. The CCD array then determines the angle and causes the rangefinder to then give a corresponding analog value to be read by microcontroller. The output of the IR sensors is analog, which is connected to the analog pin of the microcontroller. The Sharp IR Range Finder-GP2Y0A02YK0F and Dagu compound infrared sensor are used in UB swarm robot system.

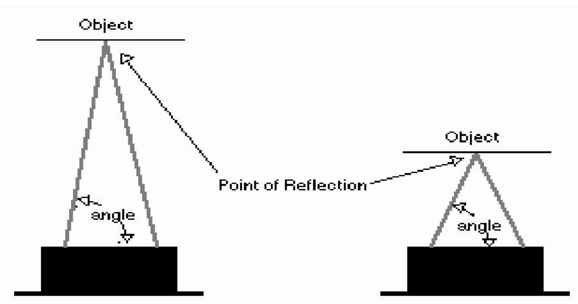

Fig. 3 – IR Triangulation Method

3.2 ENCODER

To determine the exact position or location of the robot; Odometry [14] is a more reliable, very precise technique and inexpensive. Encoder counts the number of pulses for every rotation of the wheel and from that rotation of wheel, distance can be calculated. The encoder has the IR reflective sensors which read the black and white strips on the encoder wheel. The encoder wheel is attached to the shaft and the sensor unit is mounted on the chassis. When the shaft starts rotating, the encoder wheel also rotates and the sensor board starts counting the revolutions. The encoder shown in Fig. 4 is mounted on the chassis with micro metal gear motor. This encoder has two IR reflective sensors with a phase difference of 90 degrees and the lead – lag of the waveform will decide the forward and reverse rotation of the wheel. This encoder works on 3.3 – 5 VDC voltage and the pulse output is 48 pulses per revolution.

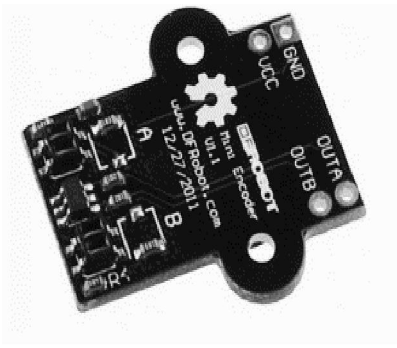

Fig. 4 – DF Robot Encoder

3.3 GPS/GPRS/GSM MODULE

Solving a task which is beyond the capability of the single robot, requires cooperation from the other swarm robots. For such a cooperative task, robots must communicate with each other and know their relative position and orientation [10]. To achieve the heterogeneity of swarm system, one of the robot uses the GPS/GPRS/GSM module shield, while other robots use encoders and vision navigation to send its relative position to the other robots as well as to the host computer. This shield with a Quad-band GSM/GPRS engine works on frequencies EGSM 900MHz/DCS 1800MHz and GSM850 MHz/PCS 1900MHz. It also supports GPS technology for satellite navigation.

3.4 CAMERA

The camera module provides vision based localization and obstacle avoidance in the swarm system. We use Blackfin Camera with Radio/Motor Board on our robot swarm.This camera can transmit the live feed to the host computer over wireless communication. In differentiating between the obstacle and goal objects, IR sensor and ultrasonic sensor have some limitations, which can be rectified by using the camera module. We can view the images on the host computer or we can also feed them to the microcontroller with the onboard image processing unit. This camera is mounted on the SRV1 platform and DF robot rover platform.

3.5 Humidity and Temperature Sensor

We are using fully calibrated digital SHT1 humidity and temperature sensor mounted on small PCB, integrated with signal processing unit. The sensor uses CMOS technology which guarantees excellent reliability and long term stability. The two wire serial interface and internal voltage regulation provides easy and fast integration with any microcontroller. This sensor consumes very low power and can be triggered only when needed.

4. LOCOMOTION AND MANIPULATION

The biggest challenges in developing the robot swarm is to make them mobile, fully autonomous and versatile so that they can move from one place to another over different types of terrains in an unknown environment [15]. The locomotion of a robot can be achieved by the motors with some gear ratio to slow down the speed of rotation and increase the torque. In manipulation, objects are moved from one place to another with the help of actuators as well as the use of motors to rotate the wrist or open and close the gripper to grab the objects. In our previous work [3], the locomotion and manipulation of different robot platforms is explained in detail. In this section, we explain the type of motors used and their connection and control mechanism with microcontroller. The robot swarm uses track and wheel for locomotion and for manipulation uses robot arm which are driven by the DC motors, Geared DC motors, and Servo Motors. These motors need motor controller to control their speed of rotation and the direction. The number of rotations can be measured by the encoder to determine the exact position of the robots using odometry.

4.1 MOTORS

The drive motor is selected based on the voltage, RPM, and either brushed or brushless parameters. The UB swarm robots are driven by motors which are attached to the wheels. On each robot, two motors are attached to the wheels along with encoder modules. We are using DC gear motors; Solarbotics gear motors, Micro-metal gear motors, and Tamiya gearbox motors. These motors are actuated and controlled using the motor controllers. The specification of motors use on UB swarm robots is given in Table 2.

Table 2. Specification of Motors.

Tamiya Twin-Motor	**Micro Metal Gear Motor**	**Solarbotics GM9 Gear Motor**	**Hitec HS-422 Servo Motor**
Gear ratios: 58:1	Gear ratio: 50:1	Gear ratio: 143:1	Speed: 0.16 sec
Motor RPM:12300	Motor RPM: 13000	Motor RPM: 78	Control Signal: Pulse Width Control
Voltage: 1.5-3VDC	Voltage: 6VDC	Voltage: 3-6 VDC	Voltage: 4-6 VDC

4.2 MOTOR CONTROLLER

We use the motor controller to drive the wheel motors in addition to the microcontroller. Figure 4 shows the Pololu low voltage dual motor controller which is mounted on Rover 5 to control the speed and direction of the wheel motors. This low voltage dual motor controller is specially designed for the motors that require low voltage and high current to drive. The left side motor’s positive terminal (Black wire) is connected to M0+ and negative terminal (Red wire) is connected to the M0- of the motor controller. The right side motor’s positive terminal is connected to the M1+ and negative terminal connected to the M1- on the motor controller. The Vcc terminal of motor controller is connected to the 5 V on microcontroller. The GND of the battery, motor controller and microcontroller are connected to each other. The SER pin of the motor controller is connected to the Pin 1 – Tx pin of the microcontroller and RST on motor controller is connected to the RST pin on microcontroller. The complete wiring diagram for the motor controller and microcontroller of Rover 5 is shown in Fig 5.

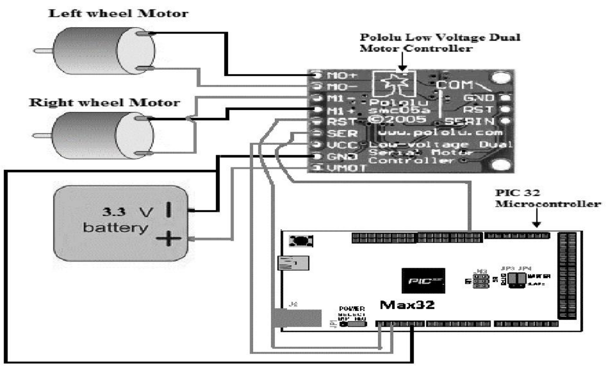

Fig. 5 – Motor Controller Wiring

4.3 MANIPULATOR WITH GRIPPER

To add more flexibility and modularity to the robot swarms, small manipulator arms with grippers are attached on the chassis. These arms are within two or three Degree of Freedom (DOF) and were built in the UB RISC lab, using the off the shelf materials such as aluminum plates, plastic materials, nut, screws etc. In theory, advanced modularity and versatility is easy to explain, but increasingly difficult to achieve and implement at the hardware level [16].

Fig. 6, shows images of the small arm with gripper mounted on robot rovers and actuated using Hitec HS-422 Servo Motors. The gripper can clasp and rotate to grab objects or to connect with other robots in the swarm. The jaws of the gripper can be opened up to 1.3” and the wrist rotates 180 degrees approximately.

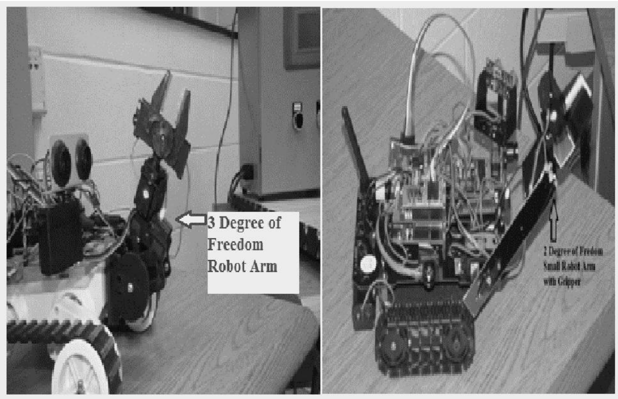

Fig. 6 – Manimpuator with Gripper.

5. COMMUNICATION AND CONTROL

5.1 COMMUNICATION

One of the most important factors for more efficient cooperative robots is the communication among them and their environment [11]. Deploying a team of robot swarms to perform specific tasks such as mapping, surveillance, pulling, rescuing, etc. requires continuous communication between the robot swarms. In our previous survey papers [3, 17], we have described all methods of communication between the robots. Communication works in different ways and it depends on factors such as communication range, environment, size of the swarm system, and type of information to be sent/received etc. In [13], the comparison between two well-known communication types – implicit and explicit has been made. The proposed robot swarm is decentralized in nature and they can communicate with each other and/or the host computer using a wireless network. Due to the advances in technology and microchip fabrication, electronic devices have become more compact and consume less power. There are many hardware devices present in the current market to accomplish the wireless communication for robot swarms. For communication, each robot swarm is equipped with X-Bee module, Bluetooth Bee module or a PmodWiFi module. X-Bee series 1, Bluetooth Bee and PmodWiFi are all compatible with each other and use same protocol for communication. The X-Bee and Bluetooth Bee use the serial transfer mode (Tx and Rx) while the PmodWiFi uses SPI mode for transmitting and receiving the data. We have created an ad hoc communication network using these modules.

The PmodWiFi module uses SPI bus as a primary interface for communicating with PIC-Max32 microcontroller on Rover 1. The SPI bus uses four signals – SS, MOSI, MISO and SCK which corresponds to the signal selection, data in/ out and clock signal. The INT provides information of data availability and data transfer complete or not to the microcontroller respectively.

5.2 CONTROL

Controlling the robot is a very difficult task, especially for a swarm system. The robots in a multi agent system are controlled using either centralized or decentralized methods [18]. The drawbacks of centralized control is explained in our previous paper [3], therefore it was decided to use a decentralized control method. If the decentralized technique is applied, the hardware structure of robots should be highly redundant with exploitation of simple and more robust control strategies. The brain for the robot is its microcontroller in which the user defined inference rules and knowledge base is stored. The performance of the robot depends on its microcontroller. The primary function of the controller is to route and manipulate the communications between other subsystems on the robot such as sensing platform, actuators, navigation system, and localization system. Robot swarms move the robots by sending the control signals to drive the motors. We use PIC32 and Arduino Uno microcontroller for our robot swarm. The programming language used for these controllers is C++ and both controllers are compatible with each other. Most of the components used on this swarm team are bought from [14]. The PIC controller is a very powerful controller, featuring a 32-bit MIPS processor core running at 80 MHz, 512K of flash program memory and 128K of SRAM data memory. In addition, the processor provides a USB 2 OTG controller, 10/100 Ethernet MAC and dual CAN controllers that can be accessed via add-on I/O shields.

Arduino Uno is an open source hardware platform, which adds flexibility in our robot swarms. This board based on the ATmega328, has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. Ultrasonic sensors as well as sharp IR sensor are connected to the analog input pins, encoders connected to the digital input pins of the controller. This board can be powered by USB port or by 3- 6VDC an external power supply. Pin 0 and Pin 1 are used for TTL serial data receiver (Rx) and data transmitter (Tx).

6. POWER CONSUMPTION AND MANAGEMENT

In the swarm robotics, the cooperation among the individual autonomous robots depends on several design parameters such as communication and management of resources [18]. The power management and distribution in swarm robotics is of very high importance, which depends not only on the electronic design but also on its mechanical structure. To perform a task in an unknown environment, robots should be capable of great degree of autonomy and operate over a longer time. The autonomous mobile robots draw power from batteries carried on the chassis in order to provide the power to the onboard sensors, actuators, and communication modules. Batteries have a limited lifetime, due to which the operational time of the robots in the swarm is also limited. For successful completion of the tasks, the robot swarm must be continuously aware of the lifetime of its power source; therefore management of power resources is necessary and vital for spending the available energy for robots swarm economically [19].

The overall power consumption can be calculated by adding the current consumed by each sensor, actuators, microcontroller and all other electronic components that are mounted on the robots. The selection of the battery depends on many factors such as size, power rating, capacity, power cycle, and cost. In the UB Swarm [20], we have five heterogeneous robots, and for each robot, we have to calculate how much power is consumed by robot. We also have to consider the other factors that affect the power consumption such as its working environment, type of terrain, elevation, how many times gripper close and pull an object. To power the UB Swarm, we have chosen Lithium Polymer batteries as a power source, which have several advantages such as high energy density, smaller size, and safe performance over the other types of batteries. In addition, these batteries have very low self-discharge rates and retention capacity. The operating current or power of each component can be found from the data sheet provided by manufacturer.

We measured the time for which sensors and actuators will be in use or active and multiply this time by their operating current, for example, if the ultrasonic sensor uses 20mA when on, and will be on 80% of the time, you get 0.8 x 20mA = 16mA.

Rover 1–

Table 3. Total Power Consumption of Rover 1.

Sr No	Component	Rating	Operating Time (%)	Current Consumption * No of Components	Total
1	Ultrasonic Sensors (SRF02)	4 mA	70 %	2.8 mA * 2	5.6 mA
2	Ultrasonic Sensors (URM V2)	20 mA	100%	20 mA*1	20 mA
3	IR Sensors (Sharp)	33 mA	50 %	16.5 mA * 1	16.5 mA
4	Temp and Humidity sensor	4 mA	10 %	0.4 mA *1	0.4 mA
5	Servos (HS 422)	120 mA	50 %	60 mA * 4	240 mA
6	Wheel Drive Motors	160 mA	100%	160 mA * 1	160 mA
7	Microcontroller (PIC)	90 mA	100%	90 mA * 1	90 mA
8	Encoders	4 mA	100%	4 mA * 2	8 mA
9	Motor Controller	10 mA	100 %	10 mA * 1	10 mA
10	Miscellaneous	100 mA	100 %	100 mA * 1	100 mA
				Total	650.5 mA

On this rover, a 2000mAh Lithium-Polymer battery is used to supply the power, and the total power consumed by this rover is 650.5 mA. So the battery lifetime can be calculated as

Battery Life = Battery Capacity / Total power consumed or required for robot

= 2000mAh/650.5mA

= 3.07 Hrs.

Rover 2–

Table 4. Total Power Consumption of Rover 2.

Sr. No.	Component	Rating	Opera-ting Time	Current Con-sumption * No of Com-ponents	Total
1	Ultrasonic Sensors (EZ1)	2 mA	70 %	1.4 mA * 4	5.6 mA
2	IR Sensors (Sharp)	33 mA	50%	16.5 mA * 1	16.5 mA
3	X - Bee	250 mA	80%	200 mA * 1	200 mA
4	Servos (HS 422)	120 mA	50%	60 mA * 2	120 mA
5	Wheel Drive Motors	250 mA	100%	250 mA * 1	250 mA
6	Microcontroller PCB (Arduino V3)	100 mA	100%	100 mA * 1	100 mA
7	Encoders	4 mA	100%	4 mA * 2	8 mA
8	Miscellaneous	150 mA	100%	150 mA * 1	150 mA
9	Ultrasonic Sensor (Seeedstudio)	15 mA	100%	15 mA * 1	15 mA
				Total	815.1 mA

On this rover, a 2200mAh Lithium-Polymer battery is used to supply the power, and the total power consumed by robot = 815.1 mA. So the battery lifetime can be calculated as

Battery Life = Battery Capacity/Total power consumed or required for robot

= 2200mAh/815.1mA

= 2.69 Hrs.

Rover 3–

Table 5. Total Power Consumption of Rover 3.

Sr. No.	Component	Rating	Opera-ting Time	Current Con-sumption * No of Com-ponents	Total
1	Ultrasonic Sensors (SRF2)	4 mA	70 %	2.8 mA * 2	5.6 mA
2	IR Sensors (Compound)	20 mA	50%	10 mA * 1	10 mA
3	Camera (Blackfin)	145 mA	80%	116 mA * 1	116 mA
4	Servos HS 422	120 mA	50%	60 mA * 3	180 mA
5	Wheel Drive Motors	73.7 mA	100%	73.3 mA * 2	146.6 mA
6	Microcontroller (Uno)	50 mA	100 %	50 mA * 1	50 mA
7	Ultrasonic Sensor (Ping)	20 mA	100%	20 mA * 1	20 mA
8	GPS/GPRS	100 mA	80%	36 mA * 2	72 mA
9	Laser Range Finder	40 mA	90 %	100 mA * 1	70 mA
10	Miscellaneous	100 mA	100%	100 mA * 1	100 mA
				Total	770.2 mA

On this rover, a 2400mAh Lithium-Polymer battery is used to supply the power, and the total power consumed by robot = 770.2 mA. So the battery lifetime can be calculated as

Battery Life = Battery Capacity/Total power consumed or required for robot

= 2400mAh/770.2mA

= 3.11 Hrs.

Rover 4–

Table 6. Total Power Consumption of Rover 4.

Sr. No	Component	Rating	Opera-ting Time	Current Con-sumption * No of Components	Total
1	Ultrasonic Sensor (MaxSonar)	3.1 mA	80%	2.48 mA * 2	4.96 mA
2	IR Sensors (Sharp)	33 mA	50%	16.5 mA * 1	16.5 mA
3	Camera (Blackfin)	145 mA	80%	116 mA * 1	116 mA
4	Servos (HS 422)	120 mA	70%	84 mA * 1	84 mA
5	Wheel Drive Motors	100 mA	100 %	100 mA * 2	200 mA
6	Microcontroller Uno	50 mA	100 %	50 mA * 1	50 mA
7	Encoder	20 mA	100%	20 mA * 2	40 mA
8	Laser Range Finder	40 mA	90%	36 mA * 2	72 mA
9	X-Bee	250 mA	80%	200 mA * 1	200 mA
10	Miscellaneous	100 mA	100 %	100 mA * 1	100 mA
				Total	883.46 mA

On this rover, a 2000mAh Lithium-Polymer battery is used to supply the power, and the total power consumed by robot = 883.46 mA. So the battery lifetime can be calculated as

Battery Life = Battery Capacity/Total power consumed or required for robot

= 2000mAh/883.46mA = 2.2 Hrs.

Rover 5–

Table 7. Total Power Consumption of Rover 5.

Sr. No.	Component	Rating	Opera-ting Time	Current Consump-tion * No of Com-ponents	Total
1	Ultrasonic Sensors	4 mA	70 %	2.8 mA * 2	5.6 mA
2	IR Sensors (Sharp)	33 mA	50%	16.5 mA * 1	16.5 mA
3	Servos	120 mA	70%	84 mA * 1	84 mA
4	Wheel Drive Motors	100 mA	100 %	100 mA * 2	200 mA
5	Microcontroller Uno	50 mA	100 %	50 mA * 1	50 mA
6	Encoders	20 mA	100%	20 mA * 2	40 mA
7	X-Bee	250 mA	80%	200 mA * 1	200 mA
8	Miscellaneous	100 mA	100%	100 mA * 1	100 mA
				Total	696.1 mA

On this rover, a 2000mAh Lithium-Polymer battery is used to supply the power, and the total power consumed by robot = 696.1 mA. So the battery lifetime can be calculated as

Battery Life = Battery Capacity/Total power consumed or required for robot

= 2000mAh/696.1mA = 2.87 Hrs.

From the calculated power as shown in Tables 3, 4, 5, 6 and 7, each robot consumes between 650 mA to 900 mA, which ensures continuous operation for a minimum of at least three hours. For this experiment, we decided to take three different sets of measurements. The first set of measurement taken while the robot rover is carrying a load and in full motion. The full load means, all the sensors, actuators, communication units, and microprocessors are in 100% working mode. In the 100% working mode, the discharged rate of battery will be very fast and the robot rover will perform a task for three hours only as shown in fig. 7, 8, 9, 10, and 11 with a blue line. In the second set of measurements, the robot rover is in full motion with no load. In this experiment, only drive motors and only one sensor are in on mode while other sensors, actuators were in off mode. The discharged rate of battery is slower than the first case as shown in fig. 7, 8, 9, 10, and 11 with a red line. The robot rover performs the task longer than in the first case. To save battery power, we decided to do power management on the robot rover by choosing which sensor and actuator should be on for task completion. So in the algorithm, we control the on and off action of sensors, actuators, and drive motors depending on the task. In this power management method, sensors, actuators, and other components will be on only when needed; otherwise, they will go in sleep mode so that we can save battery power. The experimental measurements were plotted on graph as shown in Fig. 7, 8, 9, 10, and 11 with a black line. We can see from the graph that the robot performs tasks longer than the first two sets of measurements and the battery discharge rate is very slow.

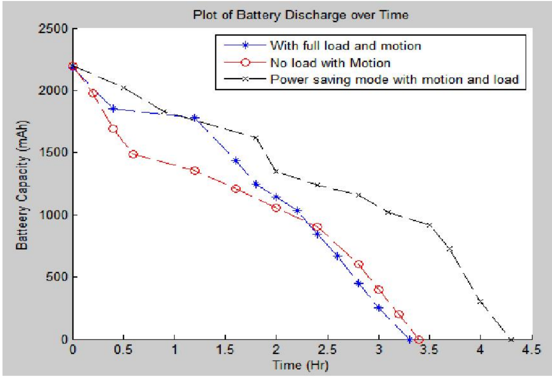

Fig. 7 – Battery Capacity Vs Operating Time for Rover 1.

For each robot of the UB swarm, current consumption is measured at different time intervals and plotted the graph in Matlab.

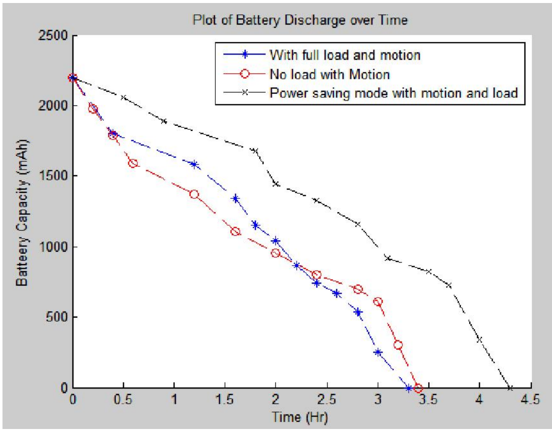

Fig. 8 –Battery Capacity Vs Operating Time for Rover 2

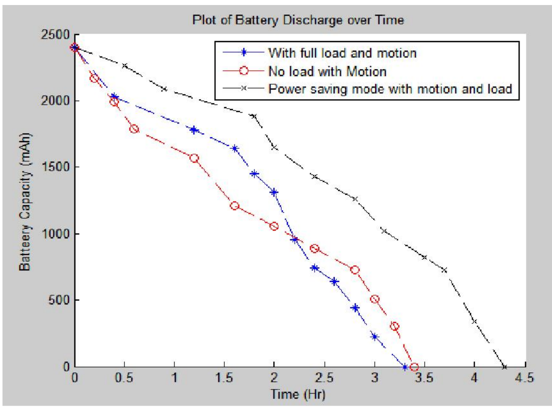

Fig. 9 –Battery Capacity Vs Operating Time for Rover 3

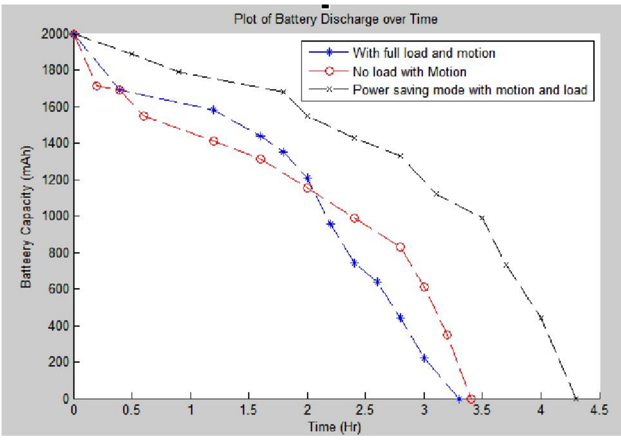

Fig. 10 –Battery Capacity Vs Operating Time for Rover 4

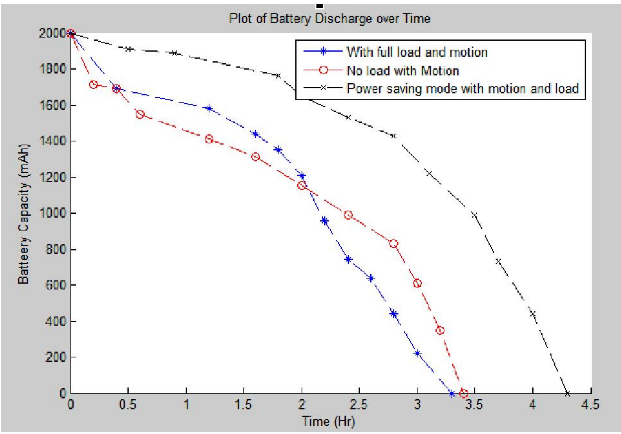

Fig. 11 –Battery Capacity Vs Operating Time for Rover 5.

The experimental measurement shows that the battery life is extended by 45 to 80 minutes by using power management technique.

7. FAULT DETECTION

A fault is a sudden, unexpected change in behavior of the robot which hampers or disturbs the normal operation of the robot in the swarm. It is essential to detect the fault in the robot swarm before focusing on the fault tolerance [19]. First we studied the types of fault that can occur in robots during a given task or in the working environment. The fault in robot swarm can occur at the physical level or at the software level. The physical level faults are related to hardware of robot such as damaged sensors, broken wheels, motors, short circuit in communication unit, while the software level faults are related with communication, algorithms as shown in Fig. 12.

Sensory data was used for fault detection to enable the robot to discover during normal operations and a probabilistic state diagram was created by using clustering technique to outline boundary limits. The isolated software component is used to monitor the data flow, and if there is change in data flow, it will give a signal to the control program. We have assigned an ID for each robot so if any fault occurs other robots in the swarm will know which robot has a fault. Following are the ID's assigned to each robot in UB swarm system: Robot1 - UB1, Robot2 – UB2, Robot3 – UB3, Robot4 – UB4, and Robot5 – UB5.

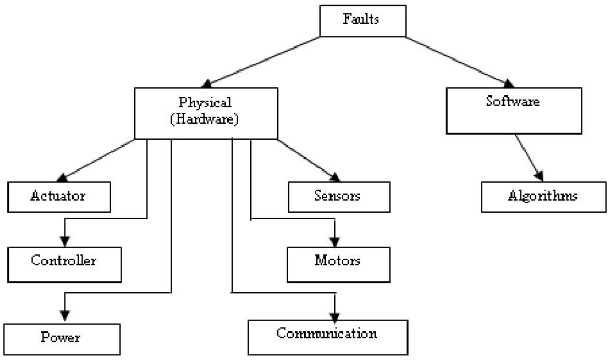

Fig. 12 – Types of Fault

We can detect the fault in wheel or drive system by using encoder readings. If we do not read or get any feedback from the encoder, then there is a fault in the wheel or motor. Fault in other sensors can be determined by checking if the input pin on the microcontroller is receiving any voltage or not. The faulty robot also sends a signal to the central system (operator) if it is in the centralized communication mode. The message signal contains the robot ID and the error code. If the other robot does not reply to robot within a certain time, there is a fault in communication unit. We have assigned tag for each fault such as given below:

F1: Sensor Failure
F2: Motor Failure
F3: Communication Failure
F4: Controller Failure
F5: Power Failure
F6: All System Failure

Whenever a fault occurred on any one of the robot of UB Swarm, that particular robot communicates to all the other robots about the fault and also central computer.

The Pseudo code for this fault detection for the micro-controller is given below,

```
if not timeout and ENQ received then
 send ACK to HostPC
else
 run robot
end if
while TRUE do
  wait for fault check
  if robot in fault then
    reply True
  else
```

```
    reply False
  end if
    check for fault
     if fault in sensor send F1 to HostPC AND other robot
       else if fault in motor send F2 to HostPC AND other robot
       else if fault in communication send F3 to HostPC AND other robot
         else if fault in controller send F4 to HostPC AND other robot
       else if fault in power send F5 to HostPC AND other robot
         else if fault in All system send F6 to HostPC AND other robot
   end if
end while
```

Fault tolerance is an ability of the swarm system to continue its operation in presence of a fault. The faulty robot or component not only affects the task completion process but also has effects on the other robots in the swarm. The fault tolerance can be achieved by hardware redundancy or software redundancy. In the hardware redundancy, we can use exactly the same type of hardware as a backup on the robot i.e. replication of the same hardware. This is a common approach for fault tolerance in sensory units. Having multiple sensory modules can act as a good fault tolerance measure. The redundant sensors can only be activated when a fault on the primary sensor is detected. If any fault occurs in any one of the sensors or components, the faulty sensor or component will be replaced by the secondary component or sensor. Adding the extra hardware will raise the other issues such as battery life, size and weight of the robot, and cost. If a motor failure, controller failure, or communication failure is detected, in such case the faulty robot will be removed from the operation or task.

8. UB SWARM

We have designed and built five UB swarm robots and performed several experiments to demonstrate the system's feasibility (video clips are available on the Web). Fig. 13 shows the images of UB swarm robots after implementing and mounting all the sensors and actuators. The hardware architecture of UB swarm robots are reconfigurable and can be reassembled at any time. The hardware architecture is also very flexible with the ability to connect any type of sensors without any modifications. This robot swarm was tested for a set of different experiments including object avoidance, object transportation, human rescue, wall painting, and mapping.

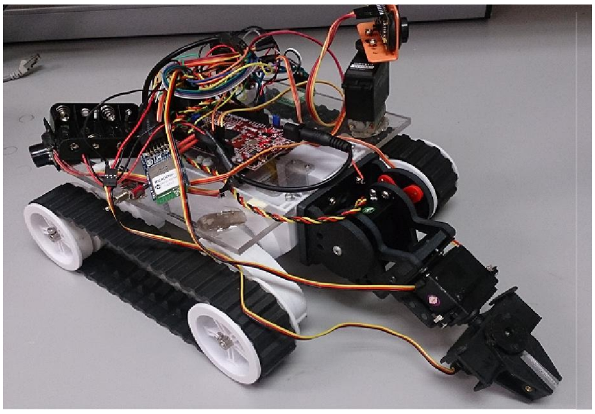

Fig. 13 – Rover 1, UB Swarm

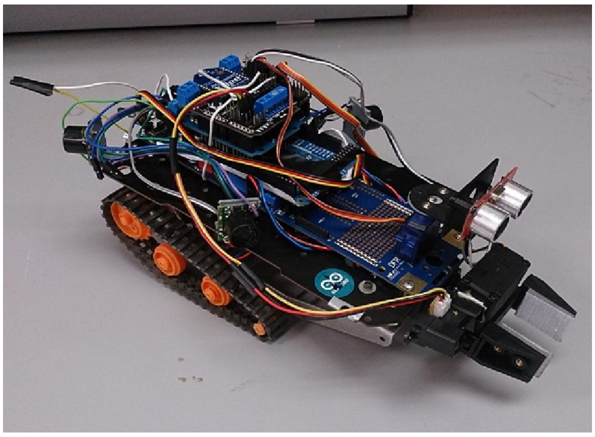

Fig. 14 –Rover 2, UB Swarm

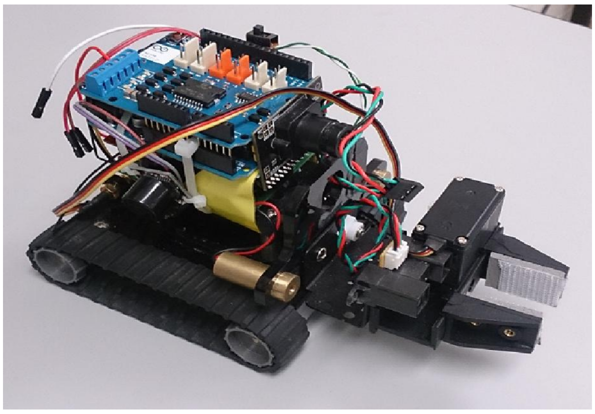

Fig. 15 –Rover 3, UB Swarm

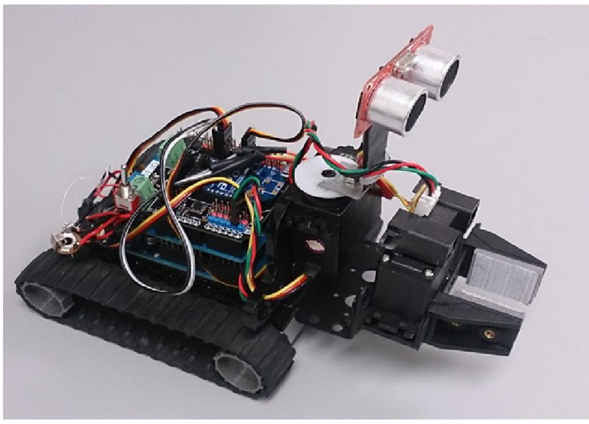

Fig. 16 –Rover 4, UB Swarm

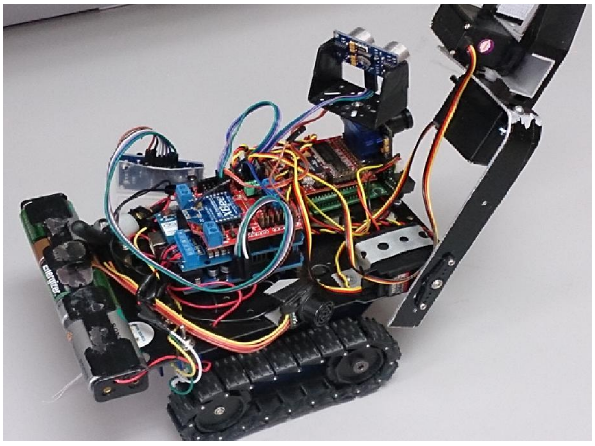

Fig. 17 –Rover 5, UB Swarm

9. DEPLOYMENT AND CONTROL

Here we also briefly describe the deployment and control architecture of the Heterogeneous RISC swarm. A detailed description and working of the software deployment environment called Robot Utility Based Task Assignment (RUTA) can be found in, our publication [26]. The deployment environment maintains a library of the heterogeneous robots that constitute the swarm along with their sensing, actuating, computing, communication and power capabilities. Two control architectures, centralized and decentralize, have been test with the swarm. The deployment environments provides a GUI where by user can add and remove robots and also various sensors and actuators.

Whenever a task is assigned to the swarm, the RUTA algorithm breaks in down into various sub-tasks that involve navigating, sensing and actuation jobs. Next, the tasks are aligned based on their timing constraints and task dependencies. Due to the heterogeneity of the robots, some robots are more suitable to perform some tasks than others. And in come case the unavailability of the sufficient battery power can render the most capable robot useless. Based on the real-time information on the battery power, sensing and actuating capabilities of the robots the RUTA algorithm selects the robots from the library by computing the utility values for the sub-tasks.

The number of robots is then optimized by resolving the task dependencies and timing constraints. In the case of centralized control of the swarm the RUTA algorithm is executed on a computer that remotely controls the swarm, while in the case of decentralized control the robots broadcast their utility values for the sub-tasks and the best robots are selected. Comparison for the cummilative utility values of the swarm for centralized and decentralized control is shown in Fig. 18. The detailed algorithms can be found in [26].

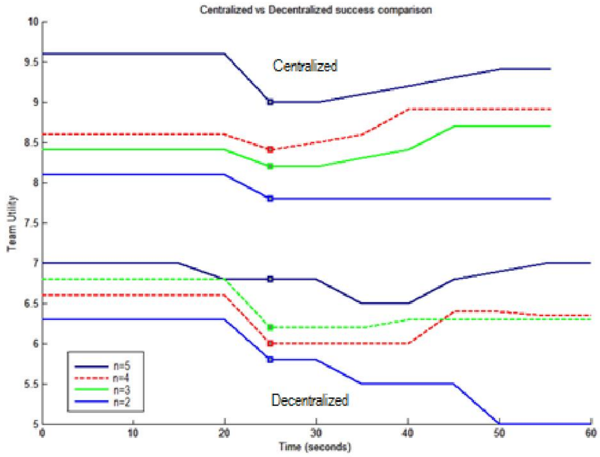

Fig. 18 – Centralized vs. Decentralized team utility

10. EXPERIMENTATION

Unstructured or unstable environments generated due to major accidents, natural disasters, and catastrophic events require urgent intervention for rescuing humans. In such situations, the common operations are search, monitoring, rescue and transport. One of the tasks we tested using our robot swarm is to rescue a human. Our demonstrated example of search and rescue task shows the different integrated abilities of these heterogeneous robot swarms including search, object detection, path planning and navigation, reconfigurability and rescue operation.

In this paper we have described a human rescue task and compare the results with increasing the number of robots in the swarm. To conduct this experiment we built small arena and initially robots placed randomly in the arena. A small web camera is mounted on the top of arena to record the experiments. We created a dummy human lying on ground inside the arena and robot swarm tries to rescue that dummy human by pulling it to a safe location. Initially we deployed only two robots of UB swarm for this task and recorded the time required by them to finish the task. After that we added one more robot to do the same task and recorded the time required for to complete. The same experimental task was replicated with deploying four and five robots of UB swarm and then comparing the time required by each to complete the task. The results of these experiments yield that the time required for five robots is much less and execution is more efficient than in the other scenarios. Fig. 19 and 20 show human being rescued by using two and four robots of UB swarm respectively.

Fig. 19 – Human Rescue using 2 UB swarm robots

Fig. 20 –Human Rescue using four UB swarm robots.

Table 8 shows the result of the human rescue task using UB robot swarm.

Table 8. Experimental result for Human Rescue.

No of Robots	Time required (Minute)	Distance travelled (feet)	Task accuracy (%)
2	20	89	48
3	17	129	54
4	14	176	63
5	10	210	72

Fault tolerance recover was also tested by introducing a fault into one of the robots during a task. The figure compares the cumulative swarm value utility over time for both centralized and decentralized schemes for the human rescue task. In case of a fault the sub-task that is assigned to the faulty robots is taken over by the rest of the team as a result of the reasoning algorithms executed by the two control schemes. The centralized results always have a higher utility than that of the Decentralized RUTA, because the centralized approach operates with complete information received from the robot team. Moreover, the decentralized approach's core functionality is based on the use of time-based parameters that not only requires more communication overhead amongst the robots but also increases the time slot given to the particular subtask and thus increases the robot's cost. Because all of the previous architectures execute greedy algorithm for task allocation, the solution quality of greedy optimization algorithms can be difficult to define.

11. COMPARATIVE ANALYSIS

Evaluating each architecture depend strongly on the nature of the experiment. Compared with the centralized RUTA, the decentralized RUTA provides more fault-tolerance and flexible method for forming solutions. However, it trades off its solution quality, requiring more communication overhead, power and more robot agents. Since other previously proposed algorithms were evaluated based on different experiments, a direct performance comparison is not possible without access to that hardware and software. For analytical comparison of our proposed approach with other approaches, we used their utility functions to calculate utility values for our swarm when applied to one of our experiments where all the task details are available. Fig. 21 shows a comparison of the swarm utility values of our centralized approach with some of the current approaches such as – AsyMTRe, M+, MURDOCH and ACO-Based. As seen from the figure RUTA for UBSwarm has better cumulative utility values when the swarm size is small. For larger swarms RUTA's utility value equals that of ASyMTRe is mainly due to the fact when large sizes are considered the role of the swarm optimization routine that improves the cumulative utility value by selecting the best robots for the sub-tasks is diminished.

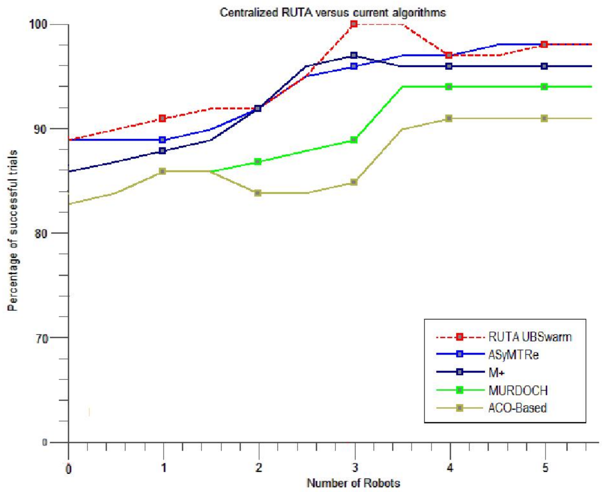

Fig. 21 –Comparison of RUTA with other approaches

12. CONCLUSION

In this work we have outlined the drawbacks of the existing swarm hardware architectures and offer new innovative techniques for more efficient systems. Most existing systems are homogeneous in nature composed of the same type robotic agents. Our survey outlines the limitation of having homogeneous swarm architecture. To overcome these limitations and add heterogeneous features to robotic swarms, we proposed novel heterogeneous hardware architecture called the UB Swarm.

UB swarm system consists of five robots which are heterogeneous in sensory units, microcontroller, functionality, and size. The proposed hardware architecture of heterogeneous robot swarm has been designed, built and tested. We describe all the hardware components used to build UB robot swarm. The power consumption and management for UB swarm with fault detection is also addressed in this work. We also present the results obtained from this work. The UB Swarm system uses both centralized and decentralized control strategies within the swarm. The robot-to-robot and robot-to-environment interaction provides the task oriented, simple collective swarm behavior.

13. REFERENCES

[1] E. Sahin, *Swarm robotics: From sources of inspiration to domains of application,* Berlin Heidelberg: Springer, 2004.

[2] L. Bayindir and E. Sahin, "A review of studies in swarm robotics," *Turkish Journal of Electrical Engineering and Computer Sciences,* Vol. 15, No. 2, pp. 115-147, 2007.

[3] A. Gautam and S. Mohan, "A review of research in multi-robot systems," in *Proceedings of the IEEE 7th International Conference on Industrial and Information Systems (ICIIS)*, 2012.

[4] M. Patil, T. Abukhalil and T. Sobh, "Hardware Architecture Review of Swarm Robotics System: Self-Reconfigurability, Self-Reassembly, and Self-Replication," *ISRN Robotics,* Vol. 2013, p. 11, 2013.

[5] S. Murata, E. Yoshida, K. Tomita and H. Kurokawa, "Hardware design of modular robotic system," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2000)*, Takamatsu, 2000.

[6] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J. L. Deneuborg and M. Dorigo, "The cooperation of swarm-bots: Physical interactions in collective robotics," *IEEE Robotics & Automation Magazine,* Vol. 12, No. 2, pp. 21-28, 2005.

[7] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. L. Christensen, A. Decugnière, G. Di Caro, F. Ducatelle, E. Ferrante, A. Förster, J. Martinez Gonzales, J. Guzzi, V. Longchamp, S. Magnenat, N. Mathews, M. Montes de Oca, R. O'Grady, C. Pinciroli, G. Pini, P. Rétornaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stützle, V. Trianni, E. Tuci, A. E. Turgut, and F. Vaussard, *Swarmanoid: a novel concept for the study of heterogeneous robotic swarms*, IRIDIA, Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle, Université Libre de Bruxelles, Technical report number TR/IRIDIA/2011-014, July 2011.

[8] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz and A. Martinoli, "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, Instituto Politécnico de Castelo Branco, 2009.

[9] M. Patil, T. Abukhalil, S. Patel and T. Sobh, "UB robot swarm – design, implementation, and power management," in *Proceedings of the 12th IEEE International Conference on Control and Automation (ICCA)*, Katmandu, 2016.

[10] D. P. Stormont, "Autonomous rescue robot swarms for first responders," in *Proceedings of the IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety*, 2005.

[11] J. Borenstein, H. R. Everett, L. Feng and D. Wehe, "Mobile robot positioning – sensors and techniques," *Journal of Robotic Systems, Special Issue on Mobile Robots*, Vol. 14, No. 4, pp. 231-249, 1996.

[12] E. Bonabeau, D. Marco and T. Guy, "Inspiration for optimization from social insect behaviour," *Nature*, Vol. 406, No. 6791, pp. 39-42, 2000.

[13] M. H. Zahugi, A. M. Shabani and T. V. Prasad, "Libot: Design of a low cost mobile robot for outdoor swarm robotics," in *Proceedings og the IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2012.

[14] Z. Yi, H. Y. Khing, C. C. Seng and Z. X. Wei, "Multi-ultrasonic sensor fusion for mobile robots," in *Proceedings of the Intelligent Vehicles Symposium*, 2000.

[15] F. Arvin, S. Khairulmizam and R. Abdul Rahman, "A short-range infrared communication for swarm mobile robots," in *Proceedings of the International Conference on Signal Processing Systems*, 2009.

[16] A. Martinelli, "The odometry error of a mobile robot with a synchronous drive system," *IEEE*

Transactions on Robotics and Automation, Vol. 18, No. 3, pp. 399-405, 2002.

[17] M. Yim, S. Wei-min, S. Behnam and R. Daniela, "Modular self-reconfigurable robot systems [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, Vol. 14, No. 1, pp. 43-52, 2007.

[18] M. Bonani, M. Stephane, R. Philippe and M.Francesco, "The hand-bot, a robot design for simultaneous climbing and manipulation," in *Proceedings of the International Conference on Intelligent Robotics and Applications*, Berlin Heidelberg, 2009.

[19] T. Abukhalil, M. Patil and T. Sobh, "A comprehensive survey on decentralized modular swarm robotic systems and deployment environments," *International Journal of Engineering (IJE),* Vol. 7, pp. 44-60, 2013.

[20] B. Ranjbar-Sahraei, F. Shabaninia, A. Nemati and S. D. Stan, "A novel robust decentralized adaptive fuzzy control for swarm formation of multiagent systems," *IEEE Transactions on Industrial Electronics,* Vol. 59, No. 8, pp. 3124-3134, 2012.

[21] A. Deshmukh, P. A. Vargas, R. Aylett and K. Brown, "Towards socially constrained power management for long-term operation of mobile robots," in *Proceedings of the 11th Conference Towards Autonomous Robotic Systems*, Plymouth, UK, 2010.

[22] A. L. Christensen, R. OGrady and M. Dorigo, "From fireflies to fault-tolerant swarms of robots," *IEEE Transactions on Evolutionary Computation,* Vol. 13, No. 4, pp. 754-766, 2009.

[23] M. Patil and T. Abukhalil, "Design and implementation of heterogeneous robot swarm," in *Proceedings of the ASEE 2014 Zone I Conference*, Bridgpeort, CT, USA., 2014.

[24] C. Churavy, M. Baker, S. Mehta, I. Pradhan, N. Scheidegger, S. Shanfelt, R. Rarick, D. Simon, "Effective implementation of a mapping swarm of robots," *IEEE Potentials*, Vol. 27, Issue 4, pp. 28-33, 2008.

[25] F. Mondada, A. Guignard, M. Bonani, D. Bar and M. Lauria, "Swarmbot: From concept to implementation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS 2003)*, 2003.

[26] T. Abukhalil, M. Patil, S. Patel, T. Sobh, "Deployment environment for a swarm of heterogeneous robots," *Robotics*, Vol. 5, No. 4, 22, 2016.

***Madhav Devidas Patil**, is currently working with Intelligrated as Control and Automation engineer. He received his PhD in Robotics and Master of Science in Electrical Engineering (Control and Automation) from University of Bridgeport, in 2014 and 2008 respectively. His research interests include Swarm Robotics, Industrial Robot, Industrial Automation, Control and Automation, Embedded systems, etc.*

***Tamer Abukhalil** received his B.sc in Computer Science from Al-al-Bayt University, Jordan. He received his MS in Computer Science from Bangalore University, India, and his PhD in Computer Science and Engineering from the University of Bridgeport. He is currently an assistant professor at the School of Information Technology, Al-Hussien Bin Talal University. His research interests include Artificial Intelligence, Robotic Systems, Digital Image Processing and Robotic Swarms*

***Sarosh Hosi Patel** received his BE in Electrical Engineering with honors from Osmania University, India. He received his MS in Electrical Engineering and Technology Management, and his PhD in Computer Science from the University of Bridgeport. He is currently a visiting assistant professor at the School of Engineering, University of Bridgeport. His current research interests include manipulator prototyping, inverse kinematic solutions, industrial control, and robotic swarms.*

***Tarek M. Sobh** received his BSc in Engineering degree with honours in Computer Science and Automatic Control from the Faculty of Engineering, Alexandria University, Egypt in 1988, and MS and PhD degrees in Computer and Information Science from the School of Engineering, University of Pennsylvania in 1989 and 1991, respectively. He is currently the Vice President for Graduate Studies and Research, Dean of the School of Engineering. Distinguished Professor of Engineering and Computer Science at the University of Bridgeport (UB), Connecticut; the Founding Director of the Interdisciplinary Robotics, Intelligent Sensing, and Control (RISC) laboratory; the co-Founder of the High-Tech Business Incubator at UB. He is a Fellow of the African Academy of Sciences and a member of the Connecticut Academy of Science and Engineering.*