Analysis of Algorithms to implement Swarm Bots for Effective Swarm Robotics

Deepika Rani Kampally, Rashmi Annae, Santhosh Kumar
Department of Computer Science and Engineering
Hyderabad institute of Technology and Management
Hyderabad, A.P, INDIA

Abstract— **The swarm intelligence paradigm has established to have very interesting properties such as robustness, flexibility and ability to solve complex problems exploiting parallelism and self-organization. Several robotics implementations of this paradigm confirm that these properties can be exploited for the control of a population of physically autonomous mobile robots. Swarm robotics is a new approach to the coordination of large numbers of relatively simple robots which takes its inspiration from social insects. This paper analyzes different algorithms that are designed for the working of a swarm robot and how they enable the multiple physical quadrupedal robots to diagnose and recover when placed in unanticipated situations. The algorithms dealt in this paper are explained on the basis of two projects where in the technical and practical aspects are examined based on the theoretical approach.**

Keywords: swarm robots, IROBOT,control,borders, Distributed.

I. INTRODUCTION

Swarm robotics is a new field, which is focused on controlling large scale homogenous multi robot systems. These Systems are made of modules that are simplified and compact in terms of design and size. These properties allow robot swarms to range from a dozen modules to a hundred. The research of swarm robotics is based on the theme of simplicity and elegance that resonates in both the designs and algorithms devised for the systems of the robots. The idea that complex macro level behaviors can emerge from simple local interactions between the agents is what the algorithms are based on. The inspiration of this paradigm is from the observations of social insects such as ants, for they are not very intelligent and don't have a centralized control, and yet they perform complex colony level behavior such as foraging of food, migrating, building of bridges etc. The complex individual robot counterparts and the combining of more numbers of robotic swarms is valuable. Robot modules are less expensive and easier to build, thus their design is straightforward. To judge the performance of the swarm robot to an individual robot is its individual entity performing complex behavior at the macro level. The obvious improvement observed is to cover more area than an individual robot. This is an analogous, for it covers different parts of a search space at once, by using the distributed search algorithm. Another improvement observed is the swarm robotics algorithms do not require the dependence of robots on each other thus the swarm robots are fault tolerant compared to an individual robot. The rest of the swarm can continue performing their actions, if a module fails, as though the module never existed, whereas if a failure occurs in a critical component of an individual robot it may become worthless. The most extremely important feature in hostile or complex environments is the robustness. Their effectiveness scales well enough with more number of members in robot swarms. To increase the effectiveness of a swarm, all that has to be done is to add, more robots. But, the improvement of the effectiveness of an individual robot is not clear, because the hardware improvement requires a software that is upgraded which is not in case of swarms. Thus, these properties of a swarm robot can make multi robot system suitable for application domains.

Although the research of swarm robotics is still rather new and has not produced a swarm of robots that has been used in a practical application. The common robot task of mapping an environment is covered through an algorithm. A swarm of robots could cover different locations at once and could disperse in an environment. The maps of the environment that are accurate and developed are super imposed on one another to provide a detailed and extensive map. Foraging is a general behavior that can be used for search and rescue or destroy, food gathering, mining organizing etc. Patrolling Has many security applications such as guarding borders , detecting intruders etc. Robot swarms cannot solve these domains. Algorithms and advancements in technology to reach these domains is discussed further in this paper.

A. Background

Ant colony optimization (ACO) is a kind of traditional swarm intelligence that does not transfer well to swarm robotics domains for several reasons. For example, in swarm robotics ,ACO[1] is hard to implement because the robots would have to alter the environment and to drop pheromones , which should be avoided as it is , as an unfavorable feature of robots systems .Dropping pheromones is a common theme when trying to transfer artificial intelligence techniques and search algorithm to robots. For example when robots use breadth first search it is extremely inefficient. A robot would have to travel to that node and backtrack instead of being able to move a pointer in a graph from node to node. Devising algorithms is itself a research for robot swarms. Swarms of robots are built or simulated to improve on the previous work, to test algorithms and to test if they work. A handful of robot swarms are built for robot swarm algorithms to serve as a platform. The research in this field is not always accurate, though they have been useful to the research community. This paper will explain and give and overview of the two largest swarm robotics projects : The IROBOT swarm and the swarm bots projects, in which the swarm bots are made manually. But we shall also discuss the self modeling and self assembling of a swarm robot using a particular algorithm.

II. ALGORITHMS FOR SWARM ROBOTICS

Implementation of a variety of algorithms Is done to run on swarms of robots. Some algorithms provide the basic functionality such as dispersion and chain formation. Even though different emergent behavior is produced by the algorithms the features remain common, among them .The basic goals of swarm robotics are derived from these features. These features include the following:

1. Simple:

The act of the individual robots is simplified using the robot controller. A state machine with a fewer edges and states represents the behavior of an individual robot.

2.Ascendable:

Swarm robotics are expected to scale well as new robots are added and are designed so that they work for any number of robots .

3. **Decentralized:**

Swarm robots are autonomous and they do not follow exterior command. A swarm member can predict as well as directly influence the behavior of another swarm bot which depends on the choice of the swarm robot.

4.Usage of local interaction:

Majority of the algorithms use local interactions over broadcasting messages and these broadcasts are used as message hopping protocols.

The IROBOT Swarm:

The massauchettes Institute of technology has housed a robot swarm of 100 units .It has been used as a platform for experiments and as a multitude of projects.

Fig 1: An irobot module

An individual module consists of a rough five inch cube and has the communications hardware, a wide array of sensors and a n human interface device. Charging stations so that the modules can autonomously dock to, included in a swarm bot.

A. Working of an irobot

1.The irobot swarm modules , use the primary software tool that is an infrared communication system called ISIS.This handles communication obstacle avoidance and localization.
2.Robots determine locations and bearings of each other, when they are in close proximity. And communication is possible.
3.The message is passed on through a gradient based multi hop messaging protocol that cause to scatter throughout the swarm.
4. Messages flow through the network topology which changes constantly, following a particular characteristic gradient.

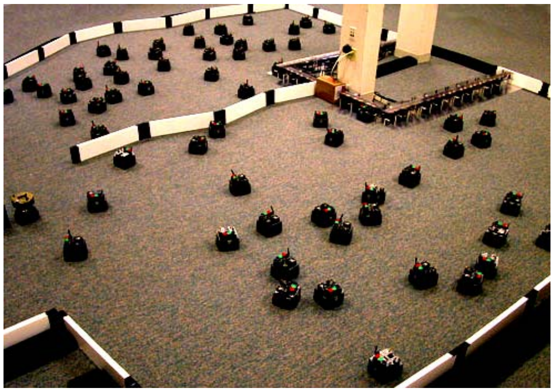


Figure 2: A iRobot Swarm

Algorithms:

The dispersion Algorithm and distributed mapping and localization algorithms are used in the working and communication of an iRobot.

The feature of the iRobot swarm, provides the researches a platform where they can implement swarm behavior at a high level of abstraction. The basic obstacle[3] avoidance and the most communication protocols are handled by the underlying system such that the emergent behavior is focused upon.

Algorithm 1:

Dispersion Algorithm :(In indoor environments):

Uniform dispersion is one of the first algorithm that has been deployed on a swarm bot. The uniform dispersion algorithm was described by Mclurkin and Smith. This algorithm is broken into two algorithms that is the uniform dispersion [4]of robots and the boundary exploration. The emergent behavior and their working in an altering fashion are provided by these algorithms. Each individual robot locates its closest Neighbors using the dispersion part of the algorithm. Distances among these neighbors are used to generate vectors to keep away from the particular neighbor. The repelling force increases when the swarmbots are close. Summation of these vectors is done and the robot intimidated about its next move .The result is that, the robots move away from one another and later it was discovered that C=2 provides the best dispersion in practical situations through empirical tests.

The next part of the algorithm that is the frontier exploration is designed to draw individual robots to the fringe robots thus the robots flow to occupy the space. During the phase each robot differentiates whether it's a frontier node (adjacent to open space) or if it's a wall node (up against the wall) or if it's an interior node (neither frontier nor wall).The gradient broadcast message is sent by the frontier nodes. A node is given in terms of hops in a breadth first manner is given from a frontier which is labeled in the broadcast message. The fastest path to the frontier is represented when the message is sent out and the interior robots accelerate towards the lowest numbered neighbor.

Fig 3: The iRobot swarm uniformly dispersed ove r somewhat complex environment (left) and an open space (right).

In a test, in a 3000 square foot schoolhouse over 100 robots dispersed in approximately 25 minutes. Due to its decentralized control algorithm the dispersion algorithm scales pretty well.

According to Mclurkin and Smith they proposed this method with a perfect example of a simple control algorithm that resulted in a complex behavior. According to their empirical results it is efficient, decentralized and scalable .This algorithm is rather complex compared to the other algorithms mentioned.

Algorithm:

```
procedure robotdecisionloop
If robot has >=1 neighbor broadcasting an alarm signal then
State <-freeze
Else if state==freeze and robot has > 1 neighbor with lower hops then
State<-scatter
Else if robot has <=1 neighbor with a lower hop count then
Robot sends alarm to neighbor with a lower hop count and lowest ID
State<- freeze
Else if robot is receiving an alarm then
State < - freeze
Else if robot is stuck then
20% chance state < - Backup
20% chance state < -Random turn
20% chance state < -Forward
20% chance state < -Random
20% chance state < -Scatter
End if
End procedure
```

Algorithm 2.

Distributed Localization and mapping Algorithm:

The exploration of a building a map for human use and most probably finding items of interest is a very common feature or task of a robot [7]. To find out, if a robot swarm is a good choice for this problem a clever algorithm is devised that generates a map and moves the swarm in a indoor environment.

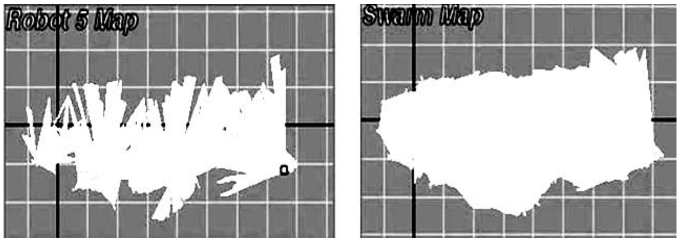


Fig 4: A visualization of the distributed mapping algorithm. The left image shows the map contribution of a single robot. The right image shows all the robots' personal maps superimposed on one another.

The constraints imposed in this aspect are that the swarm must stay together such that no communication link is lost. Thus, the only method to map a large area is to move along with a group. And in another test the robots were made to generate a map and this experiment was taken place in both restricted and open environments and the results are as per fig 3.

The major problem with the generation of a map by a robot is it must know where it is and it must have the orientation for current readings through a map sensor.

Without some sort of localization scheme, the robots are left to depend on their odometry where the localization scheme is often inaccurate and loses credibility overtime[8]. This is solved by using beacons to denote locations. It is unfavorable in the aspect of single robot situations for the environment must be modified. However the researcher Rothermich proposed that swarm robot modules can be used as mobile beacons .In an environment, as an iRobot swarm moves the anchor robots are freed in the back and formed at the front of the swarm. Thus traversal of the swarm robot is rather terrine with perfect accuracy.

Fig 5:A swarm of robots together in a group to maintain a communication link among themselves

This algorithm exhibits complex behavior which is governed by simple autonomous individual decisions made at a module level.

In this algorithm, the controller on each individual robot follows the following steps:

1. Maintaining a X,Y co ordinate ,they move in a general direction based on local beacons.
2. It starts broadcasting the position information as a beacon to the other robots if the beacons go below a certain threshold.
3. If the modules are dependent below a certain threshold, it stops being a beacon and follow the maintains a X, Y co ordinate.

Thus, this algorithm is one of the cleverest algorithms devised for a swarm robot. This algorithm appears to perform well even though multi robot mapping [15] has always been a difficult problem.

III. THE SWARM BOTS PROJECT

The swarm bots project is a large scale project based out the University of Brussels. The project has a platform of robot swarm that consists of modules called S-bots, one of which is shown in Fig 7.

Fig 6: A group S-bots joined together so they can traverse the hole.

The fig 6 shows a group [6]of s-bots joined together so they can traverse the hole between the rocks.

The s-bots use a very basic form of communication in terms of colors unlike the iRobot swarm. The swarm behavior of an S-bot can be based off on the color and the distance to other S-bots as it does not directly communicate. The robots use less cutting edge technology and are slightly smaller than the modules of an iRobot. Utilization of strong grippers to hold others to form complex structures is something that makes the S-bot stand out among other projects.

Fig 7: An individual S-bot module.

The researchers focus more on the emergent behavior rather than the complicated ad-hoc schemes, and it stays particularly to the simple design goal of swarm robotics.

Algorithms used in the swarm bot are:

1. Co-operative hole avoidance algorithm
2. Chain formation algorithm

A. Co-operative Hole Avoidance Algorithm

The first behavior implemented by any robot is moving in a n unknown environment efficiently. An S-bot group can traverse a complex landscape rather efficiently. To decrease the complexity of other algorithms implemented in a swarm bot. Trianni a researcher devised a method so that the S-bots control their own orientation, speed and direction to move about in an environment. Co operative holes [9] are hard to detect and are particularly difficult to avoid and a mistake could be fatal. A hole is detected when the robot is right on an edge of a favorable quick response. In this an algorithm, an S-bot requires two sensors which are its ground clearance sensors and its fraction sensors are used to detect movement of other S-bots. Trianni'sMethod is a devised and common approach to programming a movement controller that is the use of an evolutionary algorithm. Since the number of configurations of the S-bots could be in a vast and various type, the dynamic algorithm is required which is an evolutionary program. An evolutionary algorithm from a multi bot controller results in quick movement that does not fall into the holes[14].

The s-bots can use traction sensors to co operate accordingly and to detect where the other robots wish to go. This algorithm demonstrates a method in which a joined group of s-bots perform an important task. It uses evolutionary algorithms in a simulated environment.

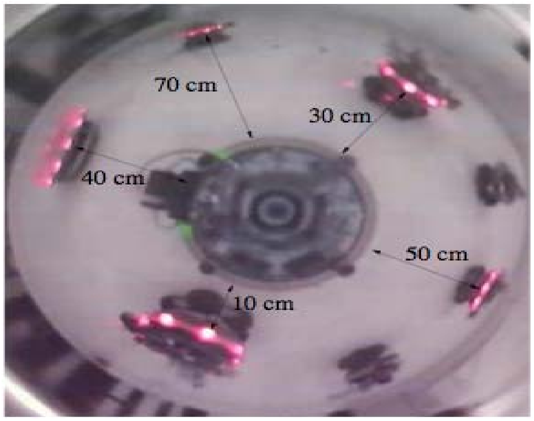


Fig 8: The view from a S-bot's omni directional camera. The distance of the object is estimated based on its size.

This algorithm is applicable for the following reasons:

1. Evolutionary algorithms are iterative
2. They maintain a large population
3. Controllers may not replicate the similar behavior.
4. The learning is transferred and simulated to the S-bots
5. Real S-bots cannot be learnt based on the new control algorithm which is the co operative hole[12] avoidance.
6. The algorithm generates in simulation a priori and can't be applied in an ad-hoc fashion.

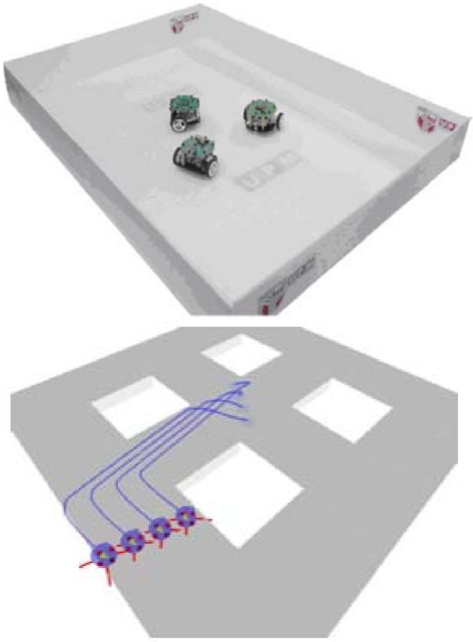


Fig 9: Four connected S-bots moving through a hole riddled environment

B. Chain based path formation

The chain based [5]path information algorithms were proposed by Nouyan and dorigo, for the swarm bots project that forms chains between a destination (prey) [14]and source (nest) with S-bots.

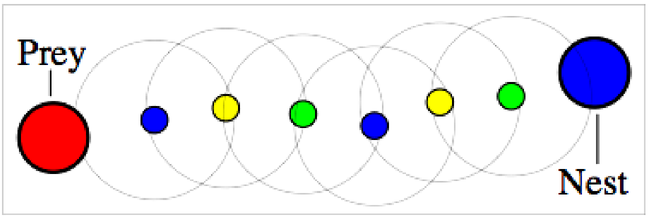


Fig 10: A visualization of S-bots

small colored circles forming a chain from a "nest" to a "prey." Each S-bot must be able to see (range given by thin grey circle) other s-bots in the chain as shown in the above figure 10 .

The communication mechanism has rather no memory and has to stay in a limited range with one another. The constraints of the chain are the prey object and the nest object [15].

This algorithm is simple and elegant as S-bot controller was designed with straightforward sates as given below:

1. search
2. explore
3. chain
4. finished

An s-bot performs a repetitive task unit .The sensory data causes the robot from one state to another state.

Outline of states and transaction of the algorithm are as follows:

- **Search** – With infrared sensors it avoids obstacles and can walk around randomly. LED's are not illuminated because s-bot does not contribute to the location of the prey[10]. Pattern in chain always follows blue, green and yellow repeatedly. S-bot sees BGYBGYBGYBGY, moving away from the nest and an s-bot sees BYGBYGBYGB, when we move towards the nest.
- Search→Explore – only if a chain member has to be detected.
- **Chain** – A chain member finds it with the color on the basis previous S-bot in the chain.

Fig 11: A picture of S-bots forming a chain from the nest to the prey

- Explore→Chain – it joins the chain or if the tail of a chain is reached, with a probability of P e→c.
- Chain→Explore –the tail of a chain, leave s the chain with a probability of Pc→e per time.
- Finished →it can move on to other tasks, such as object transport, when the job is completed where is the algorithm is presented in the next section.
- Explore→Finished – if the destination is really close.
- Explore→Search – when no chain member is in sight it is caused by an error or fault.
- Chain→Search –if the previous neighbor is no longer detectable. It is caused by an error or fault.

The experiments conducted by nouyan and Dorigo explained the effectiveness of the emergent behavior was found with a group of five to twenty s-bots and for a probability of P e→c and Pc →e the values vary. On the contrary, there is a chance that both the probabilities exist when harder tasks are benefited.

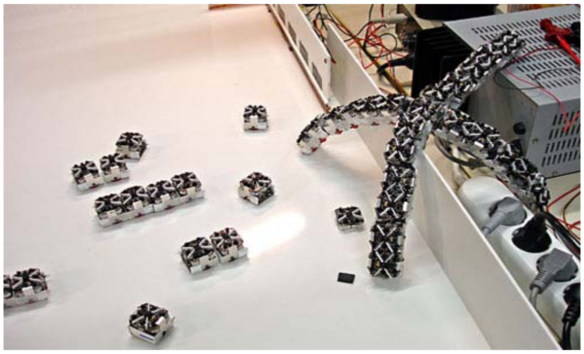


Figure 12: Group transport[6]

The swarm bot is supposed to reach its Prey and the control diagram of an S-bot of this algorithm is as mentioned in the following figure 13.

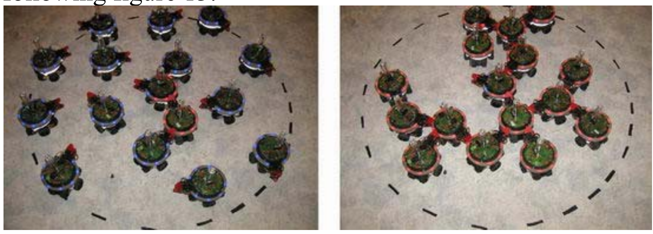

,

Figure 13: control diagram of an S-bot

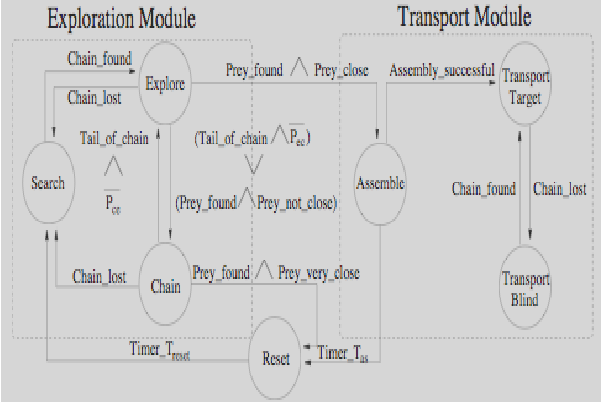


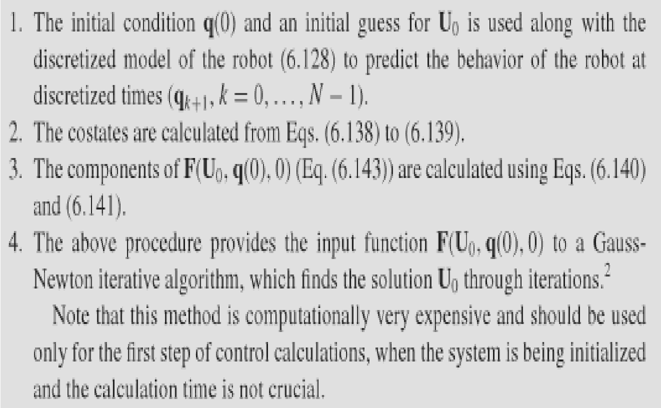
Fig 14: Control diagram in detail

The transport module is a slightly modified and augmented expression of Fig 14.The torque sensor replaces the normal sensors in the gripper of the s-bot with whose help the motion is co ordinate. The four new states are added, together which forms the Transport Module. They are outlined by Nouyan as follows:

- Assemble – it attaches to the target object
- Explore→Assemble – A red object is observed .
- Transport-Target – orientation of itself with the closest chain member.
- Transport-Blind – sense the torque on the grippers and calculate the direction to push.
- Assemble→Transport-Target – the s-bot successfully grasped onto its red target.
- Transport-Target-→Transport-Blind – the S-bot no longer detects a chain member.
- Transport-Blind→Transport-Target – the S-bot now detects a chain member in proximity.
- Reset→ something bad happened so do nothing for a specified amount of time.
- Assemble→Reset – the S-bot does not succeed in connecting to the target object.

Thus, the algorithm reached its goal in bringing the prey to the nest waiting for the idealistic view of the swarm robotics.

At the first step of control at time zero (t=0) , Uo is used to provide the initial condition.
The procedure is shown in the following Fig 15 :

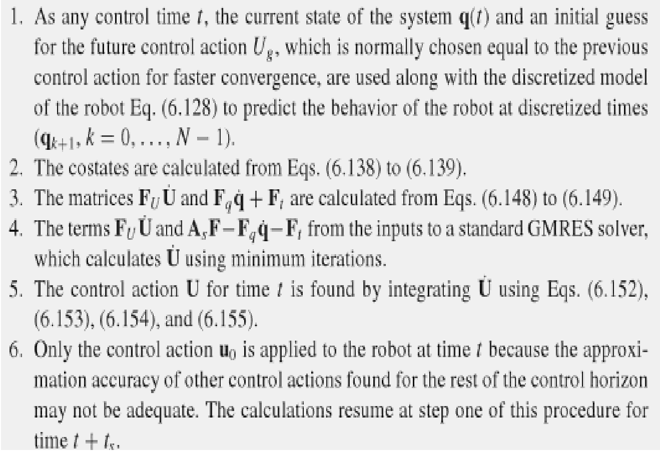


1. The initial condition $\mathbf{q}(0)$ and an initial guess for $\mathbf{U}_0$ is used along with the discretized model of the robot (6.128) to predict the behavior of the robot at discretized times $(\mathbf{q}_{k+1}, k = 0, \ldots, N-1)$.
2. The costates are calculated from Eqs. (6.138) to (6.139).
3. The components of $\mathbf{F}(\mathbf{U}_0, \mathbf{q}(0), 0)$ (Eq. (6.143)) are calculated using Eqs. (6.140) and (6.141).
4. The above procedure provides the input function $\mathbf{F}(\mathbf{U}_0, \mathbf{q}(0), 0)$ to a Gauss-Newton iterative algorithm, which finds the solution $\mathbf{U}_0$ through iterations.^{2}

   Note that this method is computationally very expensive and should be used only for the first step of control calculations, when the system is being initialized and the calculation time is not crucial.

Fig. 15:Control Module

After the initial control command Uo is calculated and the integration for U in real time is done as shown in Figure16:



1. As any control time t, the current state of the system $\mathbf{q}(t)$ and an initial guess for the future control action U_g, which is normally chosen equal to the previous control action for faster convergence, are used along with the discretized model of the robot Eq. (6.128) to predict the behavior of the robot at discretized times $(\mathbf{q}_{k+1}, k = 0, \ldots, N-1)$.
2. The costates are calculated from Eqs. (6.138) to (6.139).
3. The matrices $\mathbf{F}_U\dot{\mathbf{U}}$ and $\mathbf{F}_q\dot{\mathbf{q}} + \mathbf{F}_t$ are calculated from Eqs. (6.148) to (6.149).
4. The terms $\mathbf{F}_U\dot{\mathbf{U}}$ and $\mathbf{A}_s\mathbf{F} - \mathbf{F}_q\dot{\mathbf{q}} - \mathbf{F}_t$ from the inputs to a standard GMRES solver, which calculates $\dot{\mathbf{U}}$ using minimum iterations.
5. The control action $\mathbf{U}$ for time t is found by integrating $\dot{\mathbf{U}}$ using Eqs. (6.152), (6.153), (6.154), and (6.155).
6. Only the control action $\mathbf{u}_0$ is applied to the robot at time t because the approximation accuracy of other control actions found for the rest of the control horizon may not be adequate. The calculations resume at step one of this procedure for time $t + t_s$.

Fig. 16. Integration in Real time

IV. THE ESTIMATION–EXPLORATION ALGORITHM (EEA)

(EEA) was introduced, which uses an evolutionary algorithm to search for these informative training samples: a fitness function rewards candidate training data for how much model disagreement it causes. A second evolutionary algorithm optimizes a set of models against the current set of training data evaluated by the target system being modeled. The EEA can also be viewed as a type of co-evolutionary algorithm, in which models and tests alter the structure of one another's fitness landscapes.

The EEA has been applied to problems in machine learning, gene network identification, damage localization in truss structures, biomechanics , and robotics . In this paper, it is shown how multiple, independent physical robots with the same body plans can accelerate self-modeling [2] by sharing their experiences. This work builds on some preliminary work, in which the algorithm variants reported here were developed using a virtual system. This paper validates those approaches on a physical robot as shown in figure 17.

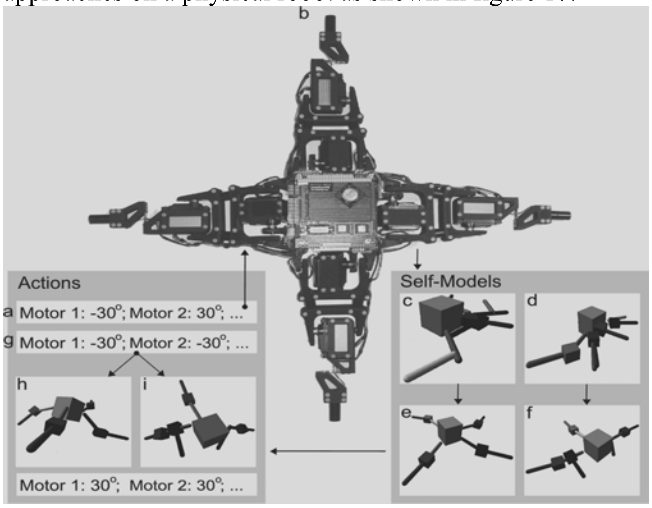


Fig. 17. Physical robot capable of autonomous self-modeling. An action is selected at random (a) and executed by the physical robot (b),which moves it from planar configuration into a static pose

The resulting orientation of the main body is recorded and along with the action that caused it, is passed to a modeling component. The modeling component then optimizes a set of self-models (c) and (d) into a new set of self-models (e) and (f) that better mimic the behavior of the physical robot[11] .A new action is then sought (g) that causes the self-models to assume maximally different poses (h) and (i) as shown in figure 18.This action is then executed on the physical robot(b). Self-modeling then recommences (c) and (d)with two action/result pairs .The process continues until a sufficiently accurate self-model is found, or a set number of cycles elapse.

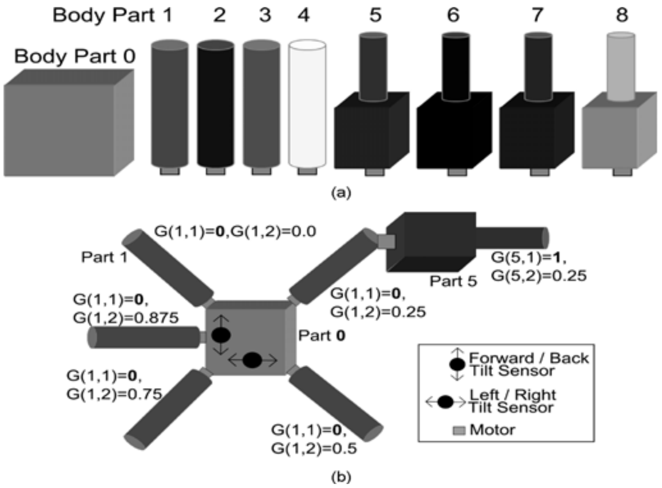


Fig.18.Genotype to phenotype translation for the self-models.

A self-model genotype encodes information for connecting the parts together indicates to which body part on the periphery it should attach. Model (b) shows five possible placements for part 1, and a sample placement for part 5.

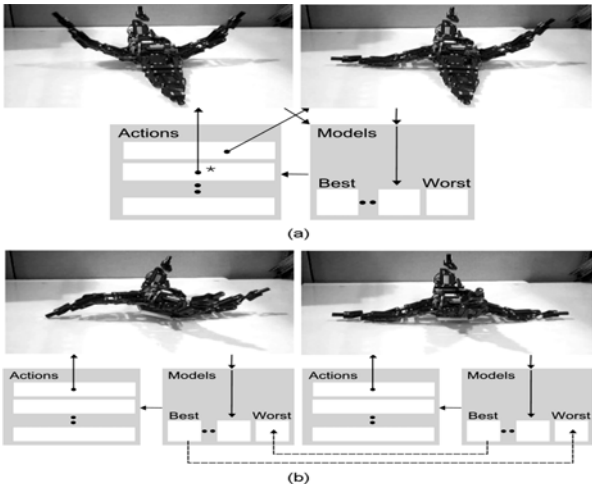


Fig. 19. Alternative approaches for exploiting two or more robots for self-modeling.

In the *Combined* variant (a), two robots each execute different actions,and then feed those two actions (along with their results) into a common base algorithmas shown in figure 19. In the *Swap* variant (b), each robot maintains its own independent base algorithm which is self configurable[13], but they swap their current best self-models.

V. CONCLUSION

In this paper we presented four interesting swarm robotics algorithms along with the self-assembling model of a swarm robot whose development and technique are advanced. The algorithms in this survey are decentralized and they completely rely on the local interactions between modules. And we suppose that advantages over individual robots is very less when compared to a group of swarm bots. And when more experiments are done we can find whether the swarm robots are actually effective or not.

ACKNOWLEDGEMENT

The authors would like to thank **Professor T. Venkat Narayana Rao**, Head and Professor, Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management[HITAM] , Hyderabad for his constant support and suggestions to improve the quality of this paper.

REFERENCES

[1]. Dorigo, M., and Di Caro, G. The ant colony optimization meta-heuristic. Mcgraw-Hill'S Advanced Topics In Computer Science Series (1999), 11–32.

[2]. Gross, R., Bonani, M., Mondada, F., and Dorigo, M. Autonomous self-assembly in mobile robotics. IEEE Transactions on Robotics (2005).

[3]. Hettiarachchi, S., and Spears, W. Moving swarm formations through obstacle fields. International Conference on Artificial Intelligence (2005).

[4]. McLurkin, J., and Smith, J. Distributed Algorithms for Dispersion in Indoor Environments using a Swarm of Autonomous Mobile Robots. 7th International Symposium on Distributed Autonomous Robotic Systems (DARS) (2004)

[5]. Nouyan, S., and Dorigo, M. Chain based path formation in swarms of robots. In ants2006, M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. St"utzle, Eds., vol. 4150 of lncs. springer-lncs, 2006, pp. 120–131.

[6]. Nouyan, S., Groß, R., Bonani, M., Mondada, F., and Dorigo, M. Group transport along a robot chain in a self-organised robot colony. In Proc. of the 9th Int. Conf. on Intelligent Autonomous Systems (2006), IOS Press, Amsterdam, The Netherlands, pp. 433–442.

[7]. Pettinaro, G., Kwee, I., Gambardella, L., Mondada, F., Floreano, D., Nolfi, S., Deneubourg, J., and Dorigo, M. Swarm Robotics: A Different Approach to Service Robotics. Proceedings of the 33rd ISR (International Symposium on Robotics) October 7 (2002),

[8]. Rothermich, J., Ecemis, M., and Gaudiano, P. Distributed localization and mapping with a robotic swarm. Swarm Robotics, Springer-Verlag (2004), 59–71.

[9]. Trianni, V., Nolfi, S., and Dorigo, M. Cooperative hole avoidance in a swarm-bot. Robotics and Autonomous Systems 54, 2 (2006), 97–103.

[10]. Aiyama, Y., Hara, M., Yabuki, T., Ota, J., & Arai, T. (1999). Cooperative transportation by two four-legged robots with implicit communication. *Robotics and Autonomous Systems*, *29*(1), Franks, N. R. (1986). Teams in social insects: Group retrieval of prey by army ants (*Eciton burchelli*, Hymenoptera: Formicidae). *Behavioral Ecology and Sociobiology*, *18*(6).

[11]. Dorigo, M., Tuci, E., Trianni, V., Groß, R., Nouyan, S., Ampatzis, C., et al. (2006). SWARM-BOT: Design and implementation of colonies of self-assembling robots.

[12]. Mataric, M. J., Nilsson, M., & Simsarian, K. T. (1995). Cooperative multi-robot box-pushing. In *Proceedings of the 1995 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (Vol. 3, p→p. 556–561). Los Alamitos, CA: IEEE Computer Society Press.

[13]. Rubenstein, M., Payne, K., Will, P., & Shen, W.-M. (2004). Docking among independent and autonomous CONRO self-reconfigurable robots. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation* (Vol. 3, pp. 2877–2882). Los Alamitos, CA: IEEE Computer Society Press.

[14]. Vakanas, G., & Krafft, B. (2001). Coordination of behavioral sequences between individuals during prey capture in a social spider, *Anelosimus eximius*. *Journal of Insect Behavior*, *14*(6), 777–798.

[15]. Wang, Z. D., Nakano, E., & Takahashi, T. (2003). Solving function distribution and behavior design problem for cooperative object handling by multiple mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, *33*(5), 537–549.

Deepika Rani Kampally, received Bachelors degree in Computer science and Engineering from JNTUH, Pursuing M.Tech in Computer Science and Engineering from JNTUH. She is having an experience of 4 Years in the field of Computer Science and Engineering, presently working as Assistant Professor in department of Computer Science and Engineering, Hyderabad Institute of Technology and Management [HITAM], Gowdavally, R.R.Dist., A.P, INDIA. Her areas of interest include network security, data mining and programming languages. E-Mail:deepikarani.d@gmail.com

Rashmi Annae, is pursuing B.Tech III year, Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management [HITAM], Gowdavally, R.R.Dist., A.P, INDIA She has organized Techno festivals (Esparto-2011) during the year 2010-11. She is member of Sahaya Society (A Social Service Organization).She can be reached at **rashmi_annae@yahoo.co.in**

K. Santhosh Kumar, is pursuing B.Tech III year , Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management [HITAM], Gowdavally, R.R.Dist., A.P, INDIA .He can be reached at **santhosh9014997365@gmail.com**